# QUERIES

IVALUA buyer

# SUMMARY


**PRESENTATION** 5

**OBJECTIVES** 7

**BECOMING FAMILIAR WITH THE INTERFACE** 7
- Accessing the Queries 7
- List of queries 7
- Query page 9
  - Query tab 9
  - Publish Tab 10
  - Query Manage tab 11
  - Action Bar 11

**CREATING AND MODIFYING QUERIES** 13

**CREATING A QUERY USING THE WIZARD** 15
- Opening the query creation form 15
- Assigning a name and a code to the query 15
- Organization of the Wizard 16
- Selecting the main table 17
- Selecting columns 18
- Filtering the query results 18
  - Applying filters before runtime 18
  - Applying filters at runtime 20
- Restricting the display of results in terms of User perimeter 20
- Organizing the display of results 21
  - Reordering columns 21
  - Sorting the results 21
  - Renaming the headers of the columns 21

**CREATING A QUERY BY COPYING AN EXISTING ONE** 22
- Duplicating an existing query 22

**WRITING YOUR OWN QUERIES IN SQL OR MDX** 23
- Entering or modifying the query's SQL code 23
  - Interactions between a modified query and the wizard 23
- Declaring the parameters included in your SQL query 23
- Parameters available without prior declaration 26
- MDX Queries 26

**DEFINING THE QUERY PUBLICATION PARAMETERS** 27
- Publish Tab – Description of fields 27
- Defining the characteristics of the output file 30
  - File name 30
  - File format 30
  - Language 30
- Linking a sub-query to a query 31

**PREVIEW AND EXECUTION** 32
- Prerequisite 32
- Previewing the raw result of the query 32
- Showing the results as a search page 33
- Running the query 33
- Viewing the logs 34
- Accessing and Managing Running Queries 35

**DISPLAYING THE RESULTS OF A QUERY** 37

**INSERTING A QUERY ON THE HOME PAGE (WEBPART)** 39
- Accessing the Configuration of the content of a default home page 39
- Adding content 40
- Selecting the query to be displayed in the webpart 41
- Quitting the Configuration mode of the Home page 41

**CONFIGURATION** **43**

**TABLE MANAGEMENT** **45**
BAS – Description of the tables of data model 45
BAS – Description of table columns 45

**MANAGING QUERY TEMPLATES** **46**
Principle 46
Creating/Modifying a *Query template* type of content 47
Multilingual templates 48

**QUERY MAXIMUM PARALLEL EXECUTION** **49**

# PRESENTATION

# Objectives

The IVALUA BUYER Queries allows the creation of SQL queries in order to retrieve information from the application database, then generate exports of results, or extract them to different file formats. Extracted information can be formatted using templates.

A wizard makes creating these queries accessible to users who have no knowledge of SQL, simply by manipulating the GUI.

Advanced users have the possibility to write their own queries directly in SQL or MDX (querying the cube). Queries also allow to create Grid format data exports that can be used within the application using the Design mode.

# Becoming familiar with the interface

## Accessing the Queries

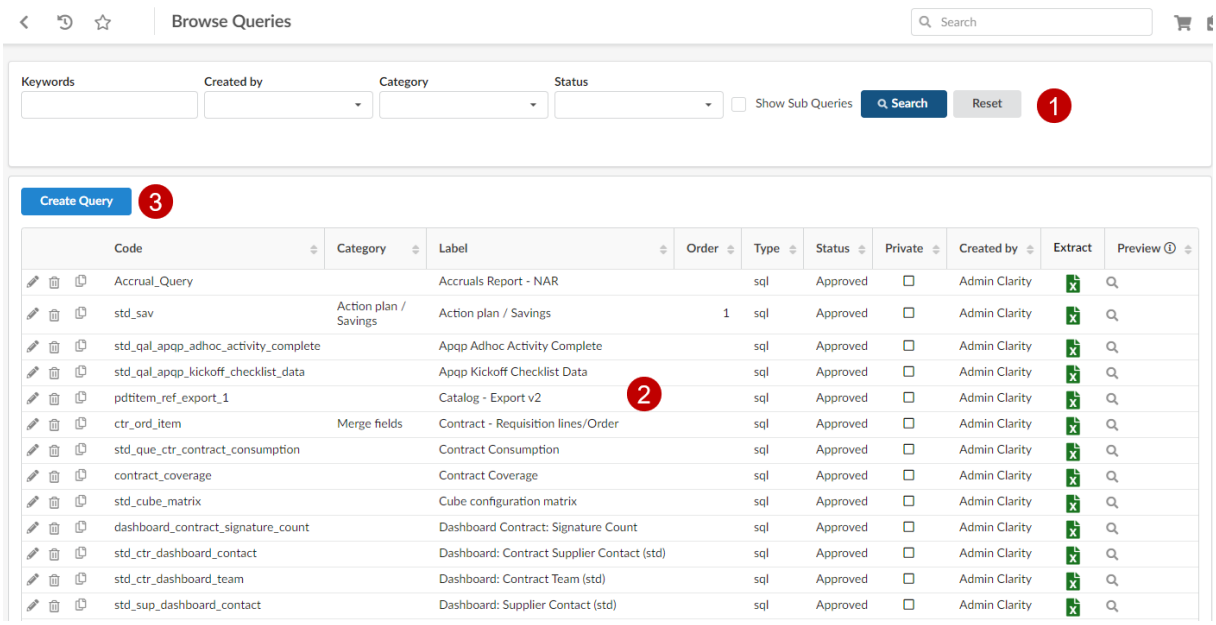
Select the *Analytics* > *Browse queries* menu. The *Queries* page is displayed.

## List of queries

The *Queries* page offers access to existing queries and allows you to create new queries.

The page is made up of the following parts:

1. The search criteria area.
2. The queries list (responding to the selected search criteria).
3. The *Create Query* button.



The query list gives the following information

| *Column* | *Description* |
|---|---|
| | Delete the query (after confirmation) |
| | Open the query form |
| | Duplicate the query (copies the query properties to create a new one based on the duplicated content). |
| Code | Unique identifier code for the query |
| Category | Group to which the query is attached (the categories are used in order to classify queries) |
| Label | Query name |
| Order | Query display order number |
| Type | The query can be one of two types:<br>▪ SQL<br>▪ MDX |
| Status | The request has one of two statuses<br>▪ Validated<br>▪ Blocked |
| Private | The query is:<br>▪ either private (only available to its creator)<br>▪ or available to all |
| Created by | Name of the user who created the query |
| Extract | Click to extract the result of the query to a file. When the query is running, the button is displayed and allows to cancel query execution. |
| Preview | Click to preview the query result in a grid (a secondary window opens) |

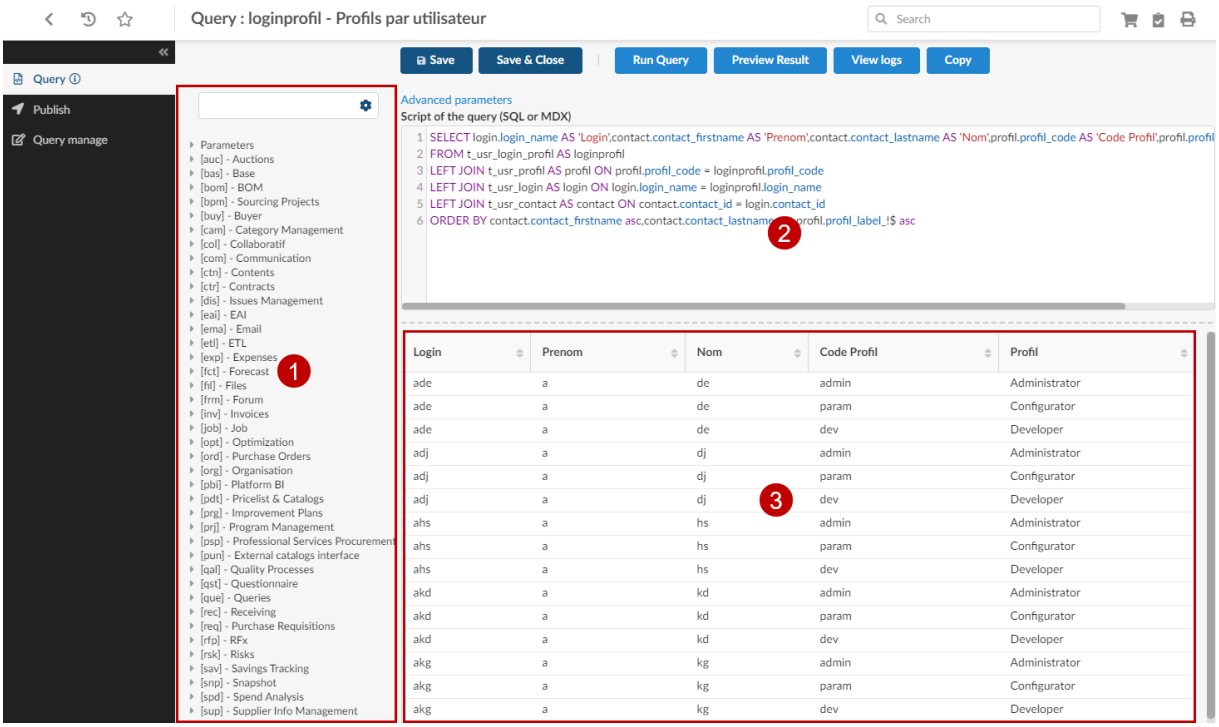
## Query page

The Query page allows manual or assisted creation of queries and the definition of publishing methods.

It is composed of 3 tabs: *Query*, *Publish* and *Query Manage*.

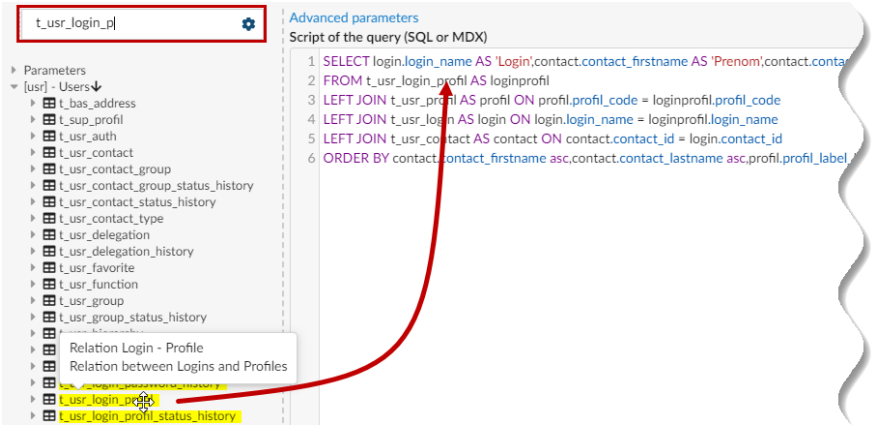
### Query tab

This tab contains the query.




The page is organized as follows:

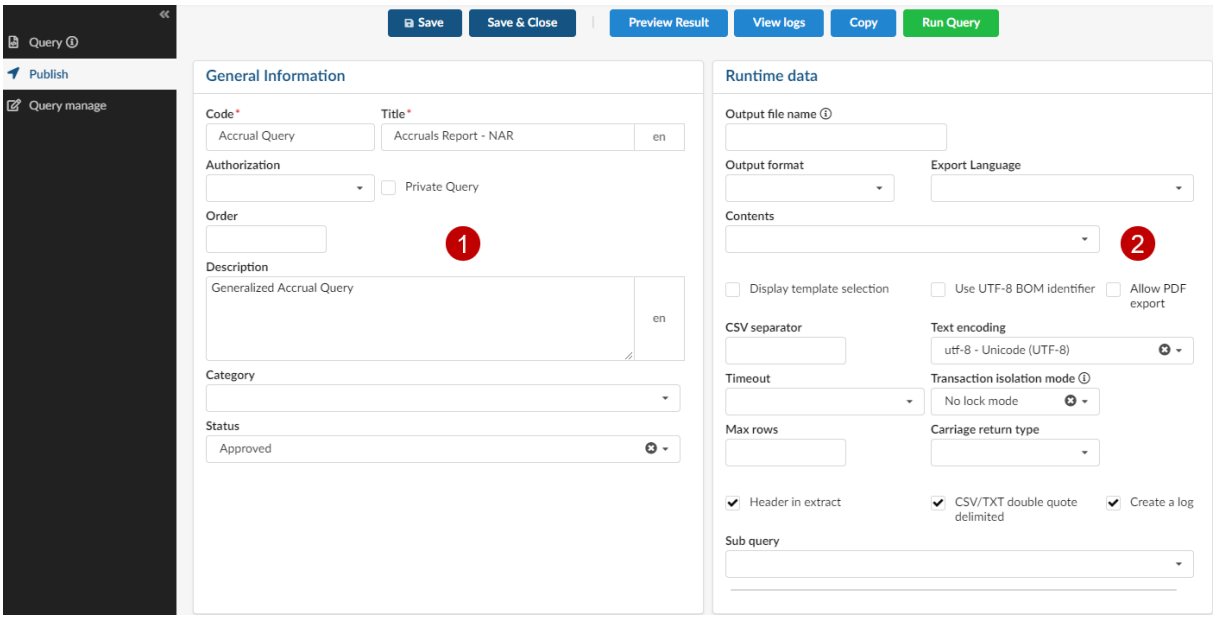
1. *Query element selection tree view*. Each node corresponds to a functional module that contains database tables. You can expand each node in the tree to access specific elements you wish to include to your query code (parameters, tables, key). You can use the search field at the top of the tree view to find a given element (matches are highlighted in yellow). The desired query element can be dragged and dropped to the position you want it to be inserted at in the code (*Script of the Query* text area).




**Note:** The Search field retrieves modules, tables and columns. The displayed list is narrowed down according to the specified keywords.

2. *Script of the Query (SQL or MDX)* area. Allows you manually write the SQL code for a query and/or change the SQL code generated by the wizard. The *Advanced parameters* link above the query code field allows you to declare parameters used in the SQL query when using direct entry instead of the wizard (see *Declaring the parameters included in your SQL query*, page 23).
3. *Query result preview*. This area displays the result of the query when clicking the *Preview Result* button. You can therefore test your SQL or MDX code safely and edit it according to the obtained query results.

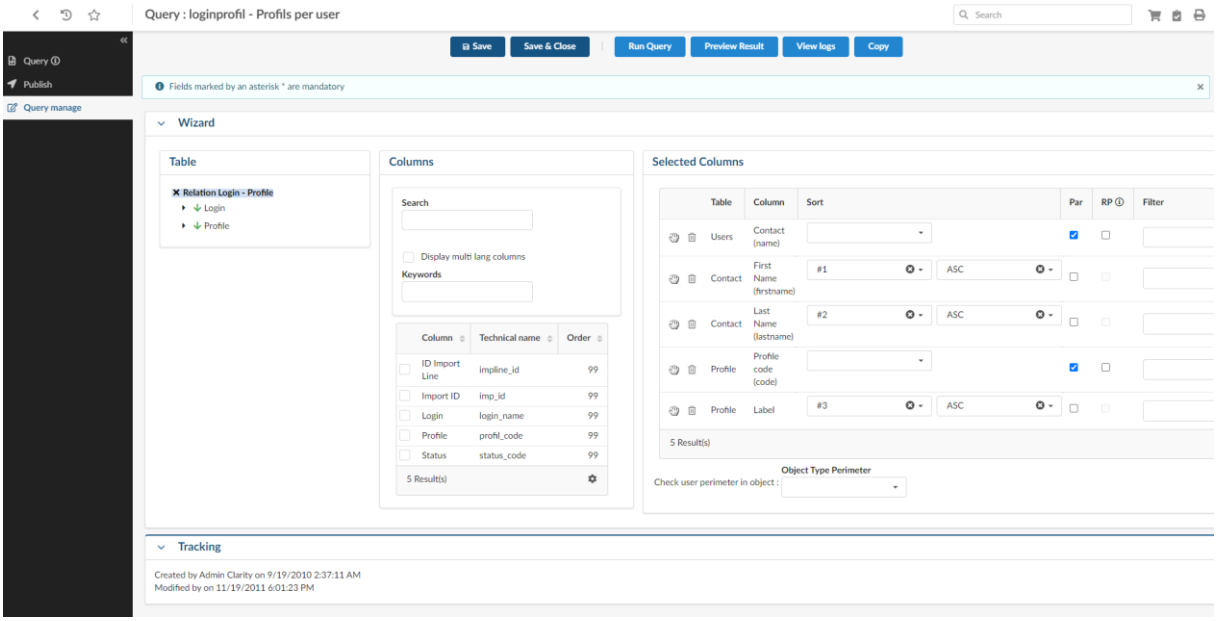
### Publish Tab




1. The *general information* area groups together the characteristics that allow the identification of the query (code, title, description) and to access it (status, authorization).
2. The *runtime data* allows the redefinition of the default characteristics of the extracted results file, to program a periodic run of the query and to link subqueries.

### Query Manage tab

This tab gives you access to the Query Wizard.




The *Wizard* area is reserved for the assisted construction of the query (no knowledge of SQL required).

The query code is automatically generated according to the tables and columns that you select, and according to the options you apply for each selected column in the wizard. The query displayed on the *Query* tab in the *Script of the query (SQL or MDX)* field is the result of the selections and configurations made on the *Query Manage* tab.

The *Tracking* area provides an account of query creation and changes over time (timestamps and name of users who have performed changes on the query).

### Action Bar

The Action Bar contains buttons that allow you to save, to preview and to run the query.

| ***Button*** | ***Description*** |
|---|---|
| Save | Saves your changes |
| Save & Close | Saves your changes and closes the query (return to the *Browse Queries* page) |
| Preview result | Displays the raw result of the query in a grid (a secondary window opens). |
| View logs | Gives you access to the query's runtime log (runtime data is logged if the option *Enable logging* is selected prior to running the query) |
| Copy | Duplicates the active query |
| Run query | Runs the query and extracts the results of the query into a file |

# CREATING AND MODIFYING QUERIES

# Creating a query using the wizard

### Opening the query creation form

Access the *Analytics* > *Browse Queries* page, then click the **Create Query** button.

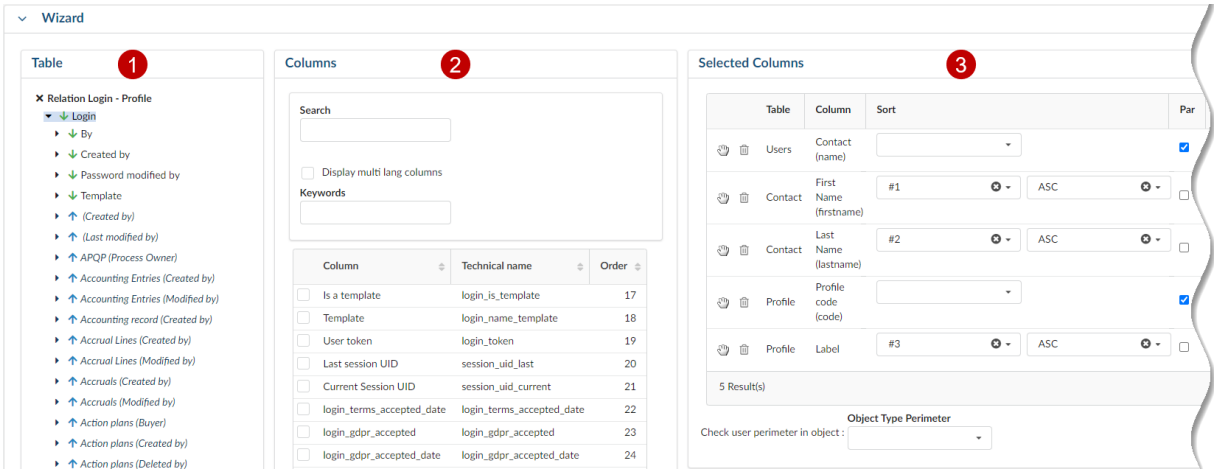
### Assigning a name and a code to the query

We recommend that you save your query from the start, which allows you to use the preview and execution features in parallel with the creation. You can therefore ensure that the query produces the expected results.

1. Access the *Publish* tab.
2. Fill the *Code* and *Name* fields.
3. Click the **Save** button.

## Organization of the Wizard

Access the *Query Manage* tab. The Wizard is made up of 3 parts:




The ***Table*** area allows the selection of the main table.

Once the selection has been made, this area shows the selected table, and the tree diagram of the tables linked to the main table (identified by the icon ↑) and the tables linked by a foreign key to a column of the main table (identified by ↓).

**Note**: To be listed in the *Main table* selector, a table must have its *Main table* (tdesc_is_main) checkbox enabled in table *BAS – Description of the tables of the data model* (t_bas_tab_desc).

The displayed tables account for the connected user's access permissions. Tto view a table, the user must be assigned the authorization set in the *Authorization* column (auth_code) of the *BAS – Description of the tables of the data model* (t_bas_tab_desc) table.

When hovering a table name, its code is displayed as a tooltip.

The ***Columns*** area displays the columns (or fields) in the order specified in the *Order* field of the selected table. The selected table could be the main table or a linked table. Checkboxes allow you to select the columns that will be used in the query. A Search field allows you to quickly locate the desired columns.

**Note**: By default, the display order of columns is determined by the order number (cdesc_order) assigned to columns in the *BAS – Description of table columns* (t_bas_col_desc) table. However, you can sort columns either by column name, technical name, or by order number by clicking the corresponding column header (click a column header once to sort on that column; click the same column header again to reverse the order).

Columns whose *Authorize the queries* attribute (cdesc_allow_query) is disabled in the *BAS – Description of table columns* table (t_bas_col_desc) are not listed, nor those of the secondary tables resulting from a foreign key.

When hovering a column name, its code and the code of the table it belongs to are displayed in a tooltip.

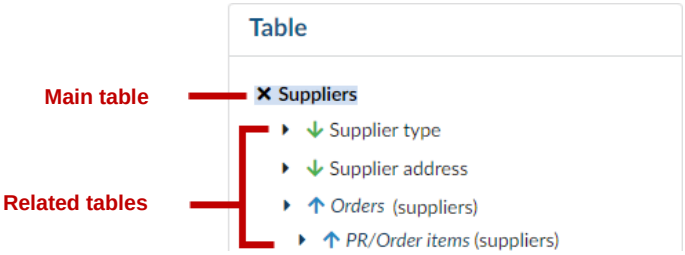
The ***Selected Columns*** area shows the list of columns selected from the *Columns* area. This area can be used to apply different functions to each column that will refine the query (parameter, filter) or modify the query output display (sorting, text columns).

**Note**: When hovering a column label, the technical name of the table it belongs to and the column technical name are displayed in a tooltip.

## Selecting the main table

1. Click the *Main table* selector button ... in the *Table* area of the Wizard.
2. The *Tables* window is displayed. Select the table that you would like to use by selecting the corresponding checkbox.

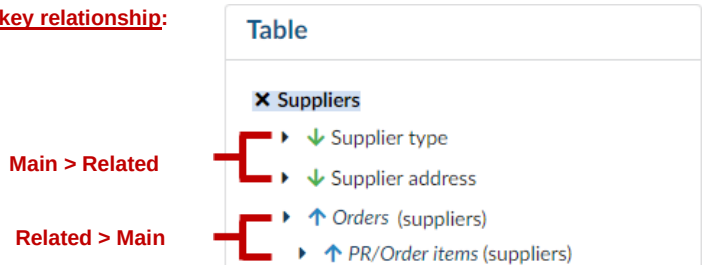
   Once done, this area will show the main table (on top) and all related tables (under the main table). The delete icon ✕ allows you to remove the main table if you want to change it (cancel the previously made main table selection).



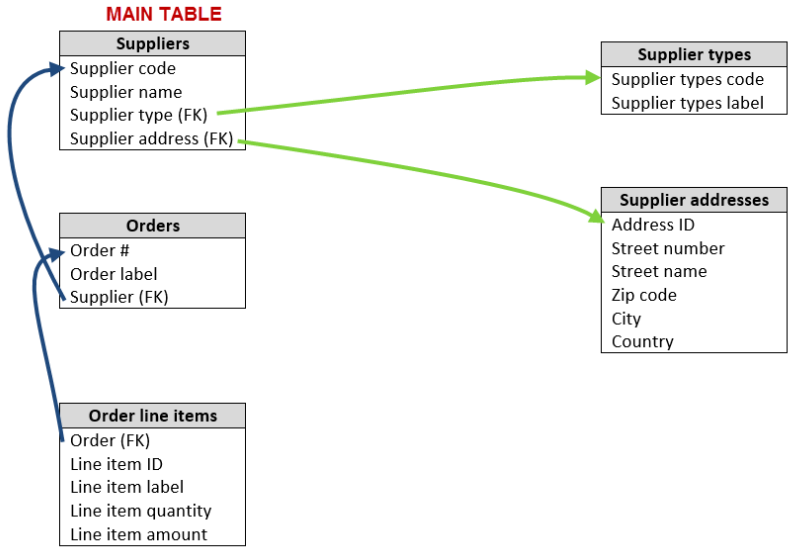

Related tables are tables that are linked to the main table by a foreign key (FK) relationship.

- If the foreign key is in the main table and points to the related table, the related table is associated with the icon ↓ (green downward pointing arrow)
- If the foreign key is in the related table and points to the main table, the related table is associated with the icon ↑ (blue upward pointing arrow)

**Foreign key relationship**:




Here is a diagram of the relationships existing between the tables listed above:




Why is this important?

When building a query, if you want to extract a label, make sure you do not select a foreign key, because you would obtain a code instead of a label. Let's take our example above: If you wish to extract the supplier type label, you must select the *Type label* column in the related *Supplier type* table, and not the *Supplier type* foreign key in the main *Supplier* table.

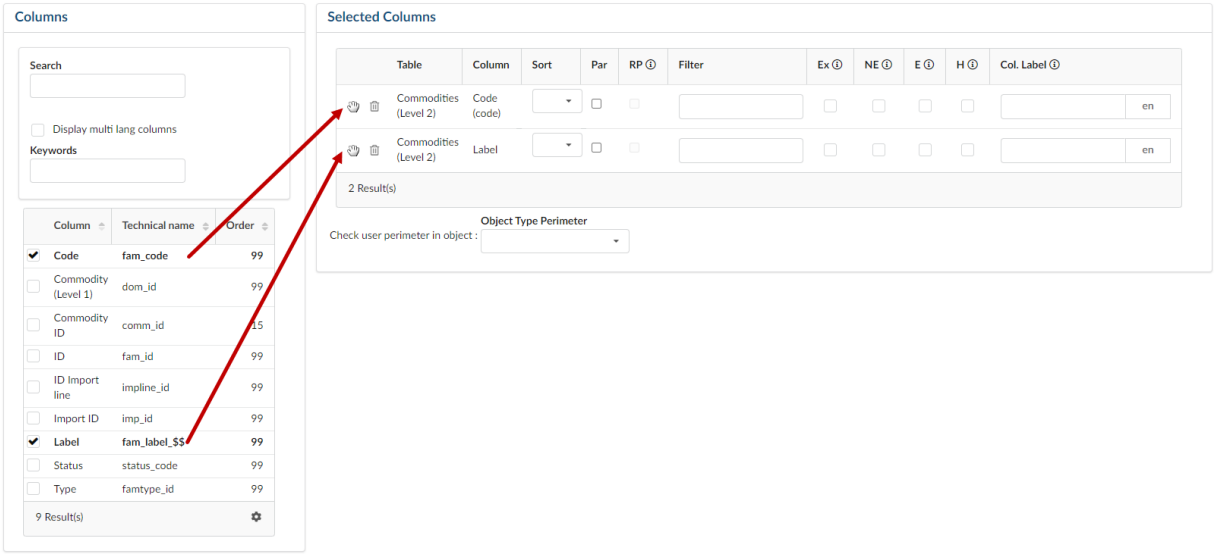
## Selecting columns

1. In the tree chart for the Main table and its dependents, select the table that contains the data that interests you.

   The list of the columns for the selected table is displayed in the *Columns* area.

2. Select the check box of the columns that you wish to use for your query. If you have a long list of available columns, you might find it handy to use the *Search* field to quickly locate the desired ones (type the first letters, then hit **Enter**).

   When you select a column, it is added to the *Selected Columns area*.

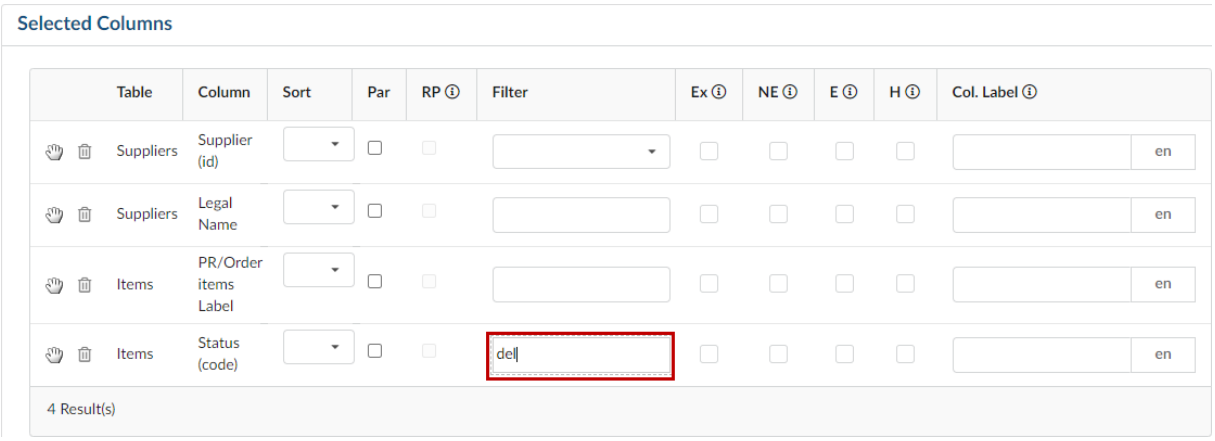



## Filtering the query results

### Applying filters before runtime

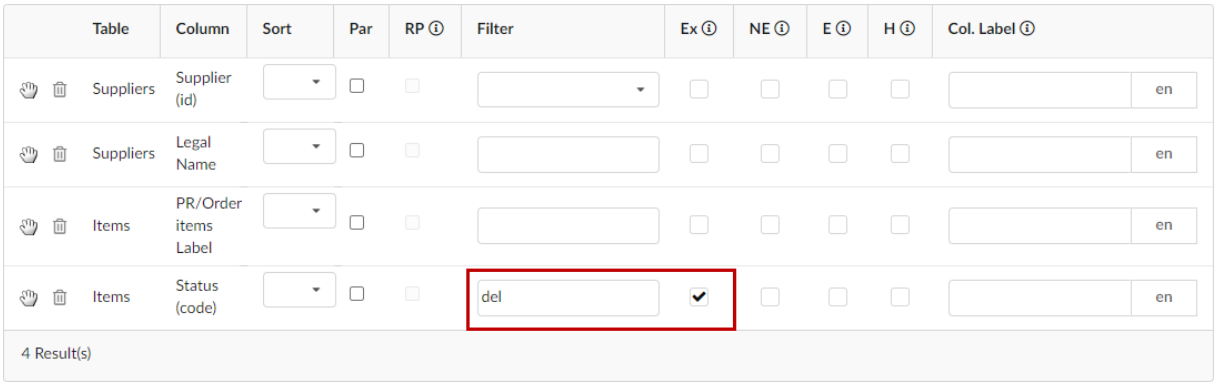
The *Filter* area allows you to enter a specific value for a given column and filter the results to this value.

In the example below, the filter on the "*del*" value of the *Status* column will list all the *Deleted* (*del*) items.




The *Exclusion* check box (Ex) inverts the effect of the *Filter* field, excluding the results of the input value (equivalent of the SQL instruction: "Not IN" or "<>").

In the example below, the filter on the "*del*" value of the *Status* column, associated with the Exclusion box (*Ex*), will list all the items that don't have the *Deleted* status (*del*).




The *Empty* (E) and *Non empty (NE)* checkboxes generate respectively "*Null*" and "*Not null*" filters. The *Hide when extract* (*H*) checkbox allows to hide the columns used to filter data in the final query extract (and in query output previews) if including specific column information is not relevant in the query output or for confidentiality purposes.

**Note:** If using a query template and the *Hide when extract (H)* option, the template configuration overrides the options selected in the Wizard. Therefore, columns that you wish to hide may still appear in the query extract (depending on the template configuration).

The form of the filter depends on the column in question.

If the column is linked to a parent table by a foreign key, 2 cases can arise:

- Either the table has a selector by default (defined in the Table management of the application) – in this case, it is this selector that appears in the *Filter* area.
- Or the table does not have a default selector and the filter takes the form of an auto-completion entry zone, associated with the *Selector* button ... .

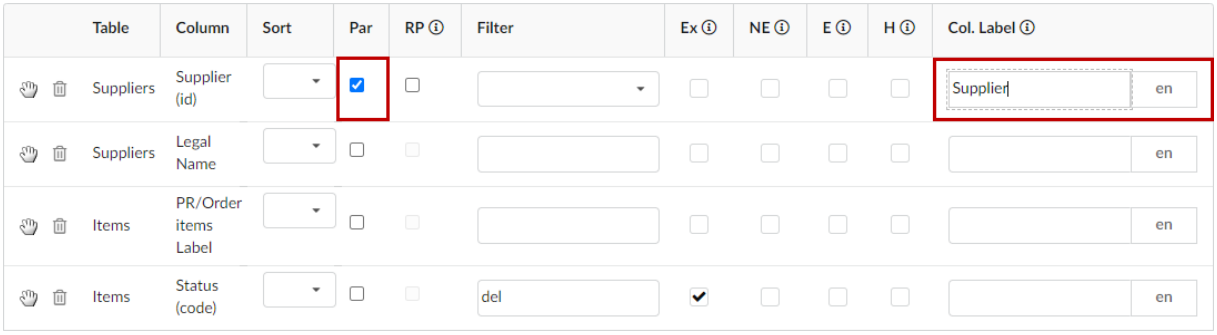
If the column is not bound by a foreign key, the filter adapts to the data format of the column in question and takes one of the following forms:

| ***Column data format*** | ***Form of filter*** |
|---|---|
| **Text**<br>Column linked to a parent table by a foreign key.<br>Contracts Contracts # | Generic auto-completion Selector |
| **Text**<br>No foreign key<br>Suppliers Legal Name | Classic text zone |
| **Numerical**<br>Contracts Amount | Dual entry zone allows the definition of a range of numbers.<br>Min value Max value |
| **Date**<br>Suppliers Creation date | Dual entry zone allows the definition of a range of dates<br>Min value Max value |
| **Boolean**<br>Suppliers Non-profit (association) | Drop-down menu proposing choices: "Empty / "Yes / No "<br>Yes<br>No |

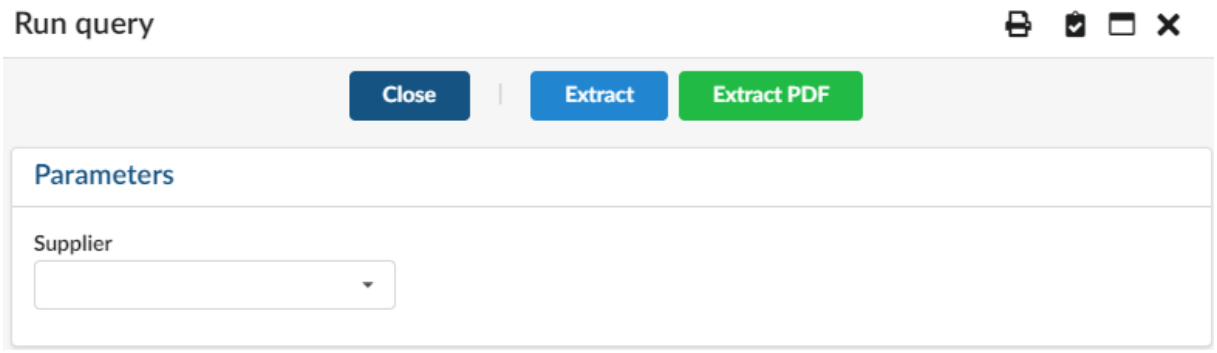
### Applying filters at runtime

The *Par* function (Parameter) allows you to display one or more filters whenever a user runs the query. The results are then filtered according to the entered filter values.

Give a label to the columns that you set as parameters so that the user who runs the query will know the purpose assigned to each filter (see *Renaming the headers of the columns*, page 21 for details).




In the example above, when running the query, the supplier selector is displayed, allowing you to select the supplier whose data you wish to extract:




You can make this parameter mandatory by selecting the *RP* checkbox (Required Parameter). Thus:

- When *RP* is disabled, users who run the query may select a filter value; however, they are not required to: they will still be able to extract the unfiltered data.
- When *RP* is enabled, a filter value must be selected in order to be able to extract the data.

**Note:** When using a Date Range type of filter (from… to…) that is generated through the wizard, the application will automatically perform a consistency check that prevents the selection of an end date that is before the start date of the range.

### Restricting the display of results in terms of User perimeter

It is possible to restrict the display of query results based on the perimeter assigned to the user by specifying the object the restriction should be applied on.

To do this, use the *Check user perimeter on object* drop down list in the *Query Manage* tab, below the selected columns.

**Note:** The access to the dropdown list is governed by the *QUE – Access to Contact Perimeter* (auth_que_contact_perimeter_access) authorization. It will not be visible to users who do not have this authorization, with the option "limited to his perimeter" applied to the main table.

## Organizing the display of results

### Reordering columns

In the *Selected column* area, each row represents a column that you have selected in the main table or in related tables.

The order of the rows determines the order of the column display in the report. The *Hand* icon allows you to reorder the columns by dragging and dropping.

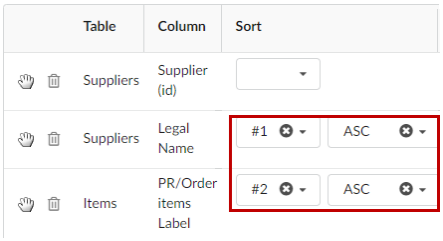
### Sorting the results

The *Sort* column allows you to organize the query results (same as the "ORDER BY" SQL instruction).

The 1st dropdown list allows you to specify the priority order in which each column is taken into account as a sorting criterion for the data.

The 2nd dropdown list (visible only when an order number has been selected) allows you to specify the sorting direction, that is, whether the results should be sorted in ascending (ASC) or descending (DESC) order.

In the following example, the results are first sorted by the supplier name, then by the item's label, in ascending order (i.e. by alphabetical order).

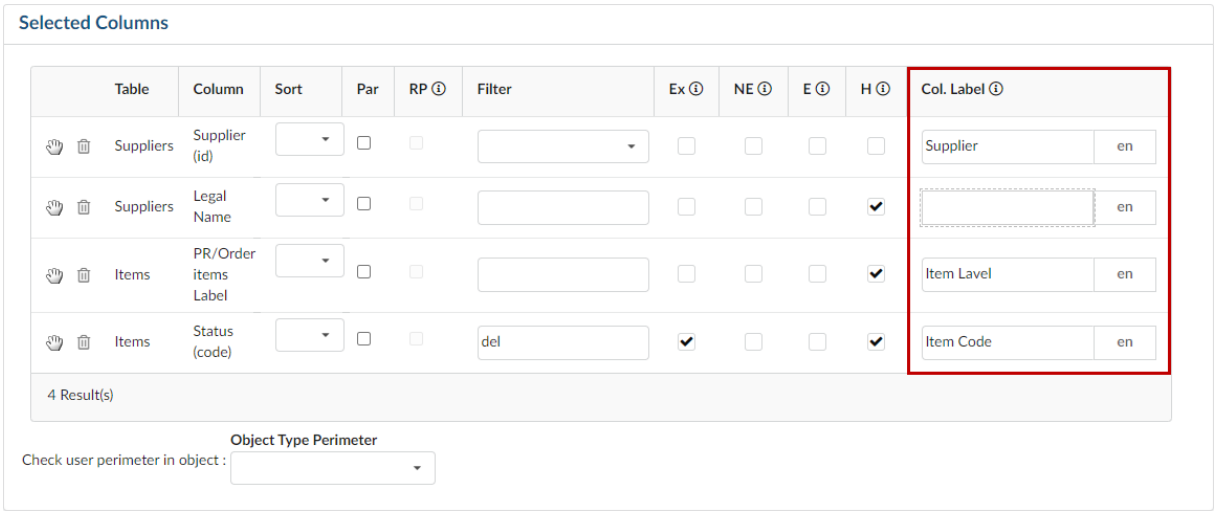



### Renaming the headers of the columns

By default, the columns in the query result correspond to the name of the columns in the database. The *Col. Label* Column (Column name) allows you to rename the columns to improve the presentation of the results.

Note that the labels entered in the *Col. Label* column will serve as aliases for the columns used as merge fields in formatted data extractions that use an Excel template (merge fields being inserted in the following form: %%query.alias, where "*query*" is the query code and "*alias*" is the field's alias in the SELECT statement of the SQL query).

The following characters are not allowed: & ' ( ) : , . + { [ | ] } ¤



# Creating a query by copying an existing one

## Duplicating an existing query

***Access***

Access to query duplication requires the authorization to edit queries (QUE - Modify a query / auth_que_query_manage).

You may duplicate a query either from the list of queries, or from the detailed sheet of the query you want to duplicate.

- From the list of queries, click the *Duplicate* icon of the query to duplicate.
- From the query you want to duplicate, click the **Copy** button in the action bar.

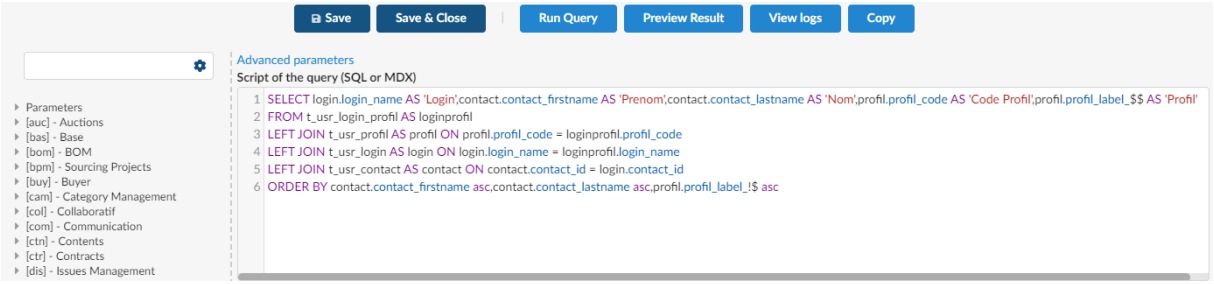
After your confirmation, a new query with the same characteristics as the copied query will display.

Its code is set by default to: *[code of copied query]*_1 (you can change this as needed).

At this stage, the new query is not saved yet; you need to give it a title and hit **Save** if you want to keep it.

# Writing your own queries in SQL or MDX

## Entering or modifying the query's SQL code




The query generated by the system is displayed in the *Script of the query (SQL or MDX)* field of the *Query* tab.

Proficient SQL users (who have the appropriate authorization) can edit the SQL code manually or write the entire query in this area without resorting to the wizard.

The query can include parameters, which need to be declared in the *Advanced parameters* window (see page 23), with the exception of a predefined set of parameters that do not require prior declaration (see page 26).

### Interactions between a modified query and the wizard

From the moment this zone is modified, the query is set to manual mode and is disassociated from the wizard. In other words: the changes made in the *wizard* are no longer displayed in the *Script of the Query* field.

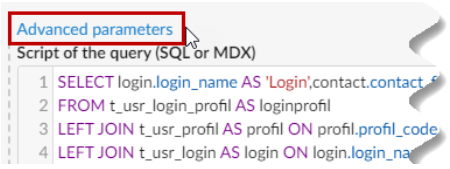
The **Build Query from Wizard** button (displayed in the action bar after the query code has been manually edited and saved) allows you to switch back to assisted Creation mode, which manually writes over the modified query that was generated by the wizard (after confirmation).

## Declaring the parameters included in your SQL query

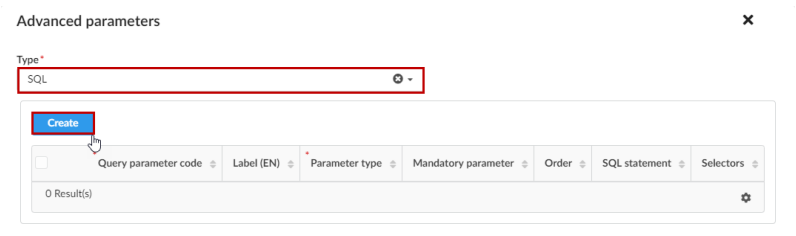
The *Advanced parameters* window allows you to declare the parameters that you are using in your SQL query in Manual mode.

To add a parameter:

1. Click the *Advanced Parameters* link located above the *Script of the query (SQL or MDX)* text field on the *Query* tab.




2. Select the *SQL* type (this enables access to the table tree view, to the SQL parameters, and set the Script of the query field to SQL mode), click the **Create** button.




The *Query parameter* window is displayed.

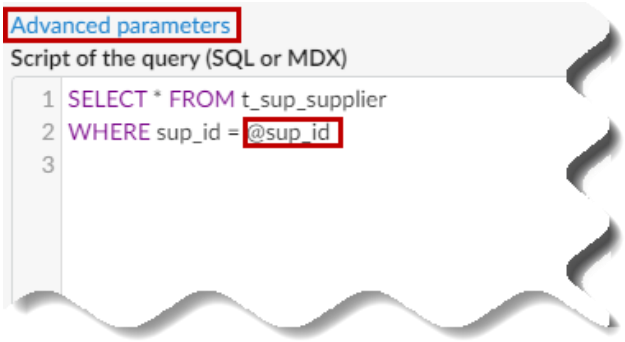
3. Fill in the fields, referring to the table below:

| ***Field*** | ***Description*** |
|---|---|
| Query parameter code | Code that identifies your query parameter uniquely. |
| Parameter Type | Appearance that the filter (parameter) will take at query execution time in order to allow the user to select a value<br>Available filter types are: Selector (single or multiple), Text (contains or exact search), Yes/No, Date, Decimal, and Integer.<br>When creating a Date Range parameter type using *Advanced Parameters*, you must then use the Design Mode to ensure the consistency of the selected dates; for instance:<br>Let's say we have a Start Date field (*date_from*) and we want to make sure the date that is selected is before that of the End Date field (*date_to*).<br>To implement this kind of check, you should access the "*date_from*" field in Design Mode and set the "*Maximum value*" as: "<=#*date_to*" |
| Label | Label of the filter as displayed at execution time (should indicate to users what they are expected to select/enter) |
| Mandatory parameter | Select this checkbox to specify that a value is mandatory. If left unselected, specifying a value is optional for this filter.<br>At runtime, mandatory parameters are signaled with a red asterisk. |
| Order | When there are multiple parameters, the order number determines the display order of parameters |
| Selectors | The selector contains the values of the filter, among which the users who run the query will be able to choose the value they want to apply |
| SQL statement | This field allows to specify how the parameter should be used in the query.<br>e.g.: If specifying *ctr.ctr_id = @ctr_id* the statement will be added to the WHERE clause of the query. |

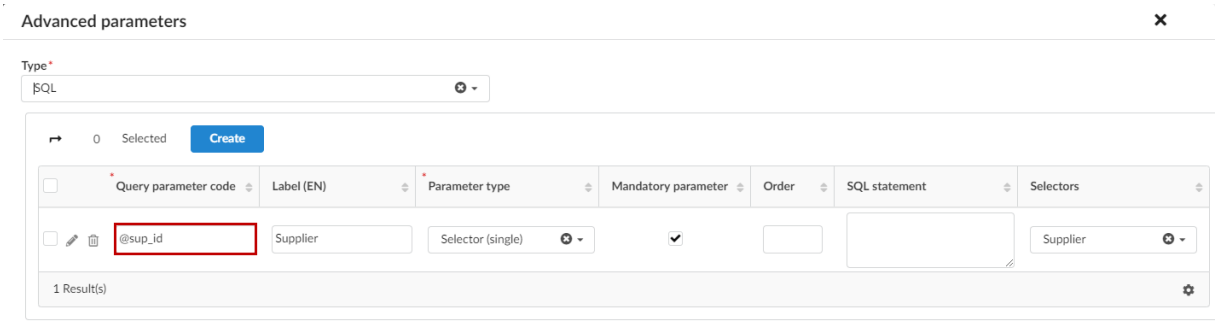
4. Click the **Save & Close** button.

Here is an example of a query that will extract the data of the selected supplier, with:

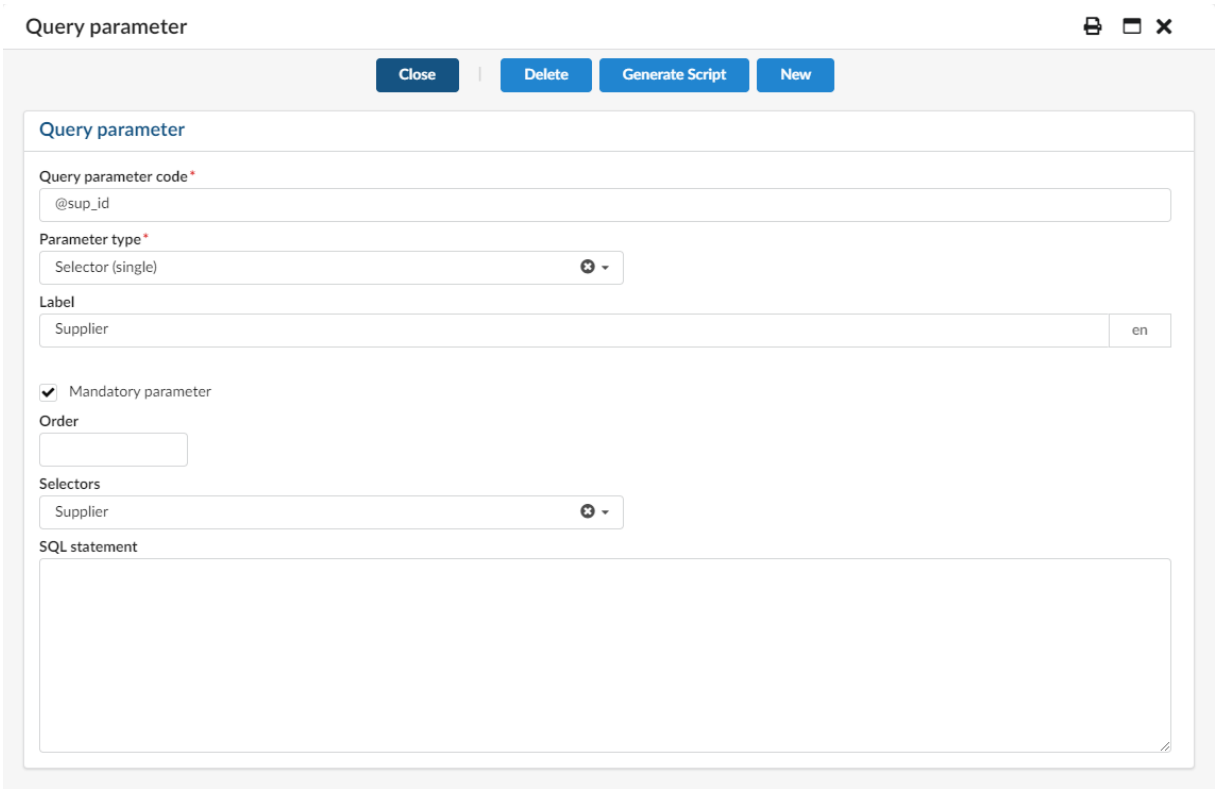
- The SQL code and the declaration of the parameter included in the query:



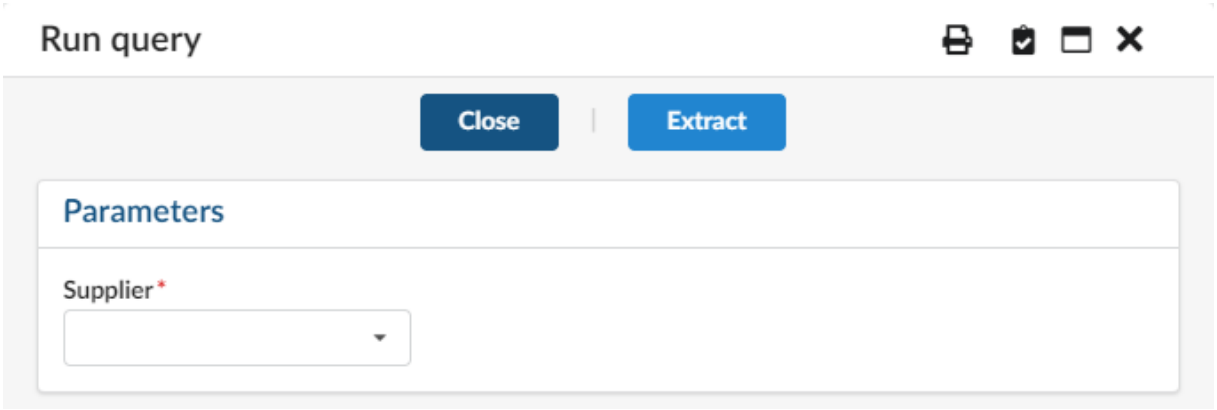
- The declared SQL parameter used in the query:




- The detail of parameter declaration:




- And the display of the parameter upon execution:




Where needed, you can assign a default value to a parameter using Design Mode.

Likewise, when the query uses multiple parameters, you can use the Design Mode to arrange filters as you wish.

### Parameters available without prior declaration

The following SQL parameters are always available within your queries and do not require prior declaration:

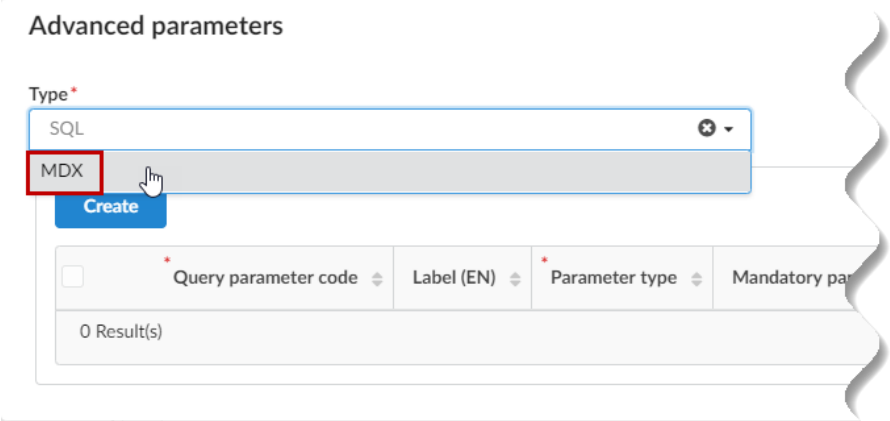
- @login_name : login ID of current user
- @base_url : WwwRoot property
- @lang : language of current user
- @url_prefix_intranet : prefix of intranet URLs
- @url_prefix_extranet : prefix of extranet URLs
- @timestamp

### MDX Queries

The *Script of the query (SQL or MDX)* field of the *Query* tab allows you to receive the MDX code (to directly query the cube).

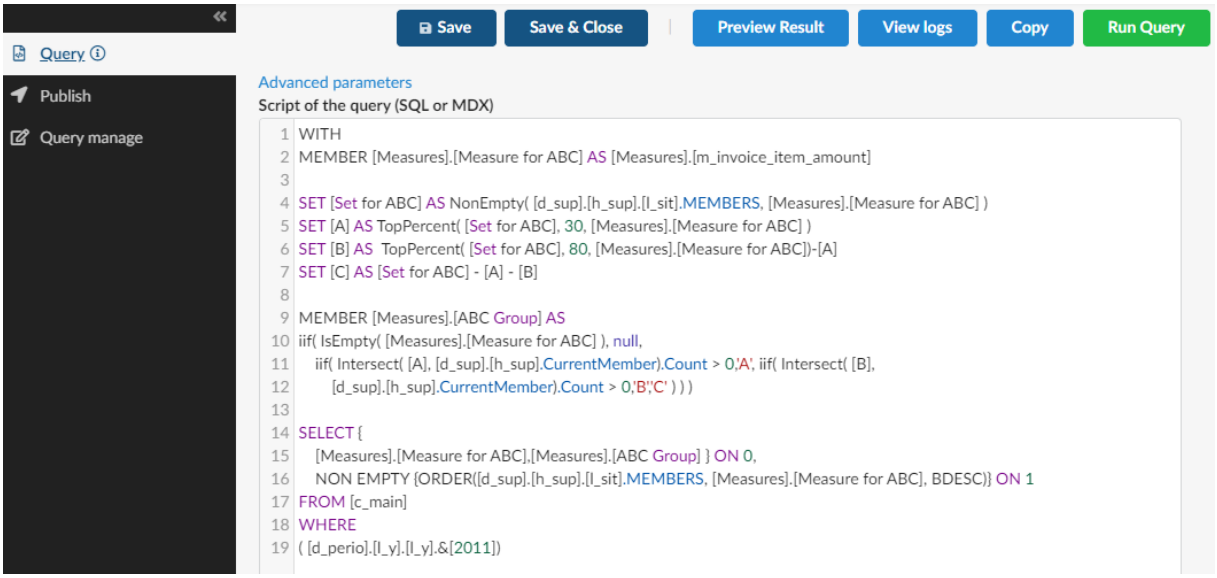
To create an MDX query, you must select the *MDX* type before writing the query code.

1. On the *Query* tab, click the *Advanced parameters* link located above the *Script of the query (SQL or MDX)* field.
2. Select the MDX type.



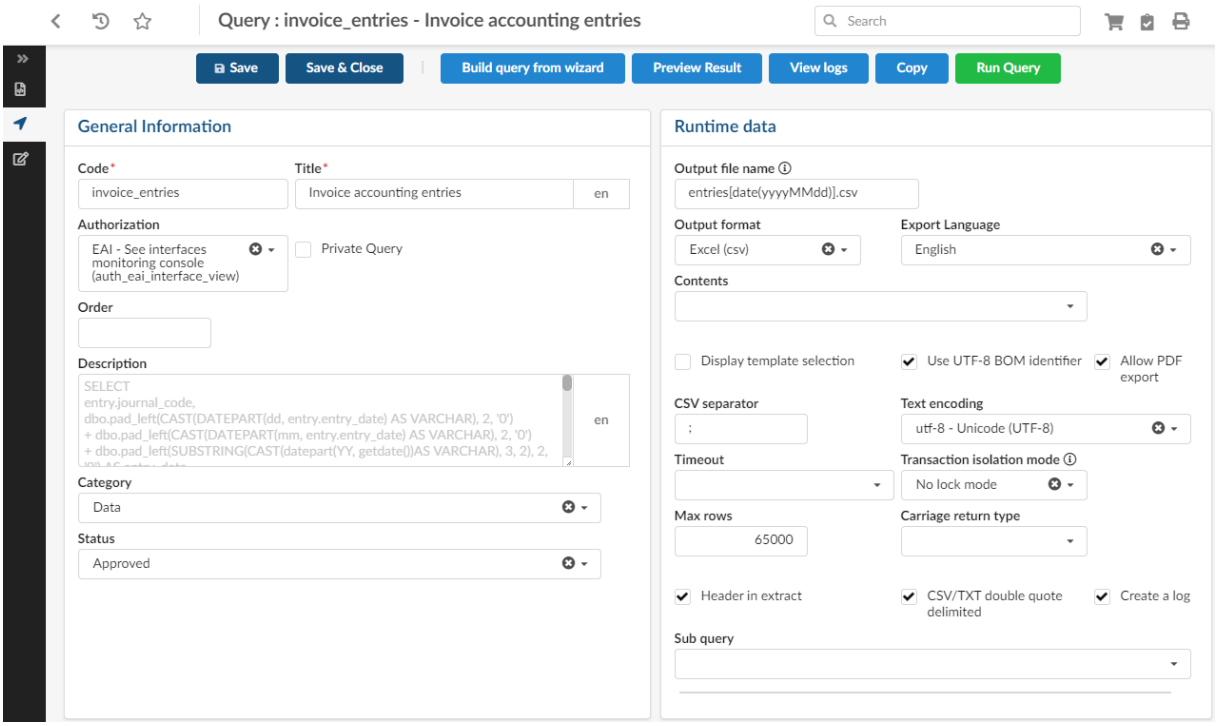

3. Close the *Advanced Parameters* window.

This will hide the *Wizard* part that cannot be used to generate MDX code.



# Defining the query publication parameters

## Publish Tab – Description of fields




The table below describes the fields of the *Publish* tab of the query.

| ***Field*** | ***Description*** |
|---|---|
| ***General Information*** | |
| Code | Unique identifier code for the query<br>By default, this field is re-entered with the alias of the main selected table. |
| Title | Query name or label |
| Category | Group to which the query is attached (the categories are used in order to classify queries) |
| Authorization | Access authorization to restrict the use of the query by authorized profiles. Note the ability to make a query private or accessible to everyone. |
| Query Type<br>*(from version 8.168)* | The query can be of two types. This allows segregating queries in order to restrict their use, therefore ensuring data integrity and confidentiality (e.g.: data shown to external or internal users, ensuring that no restricted data can be recovered by unauthorized users):<br>▪ Internal: only internal users can execute the query<br>▪ External: enforces the advanced query security validation process at creation time before allowing the query into production. External users (e.g.: supplier contacts) can run external queries only. Internal users can run external queries. |
| Order | Query display order number |
| Description | Descriptive text indicating the (for example) purpose of the query |
| Status | The query has one of two statuses |

| *Field* | *Description* |
|---|---|
| | ▪ Approved: the query can be used<br>▪ Blocked: the query cannot be used |
| ***Runtime data*** | |
| Output file name | By default, the output file is named: "*Query (n)*", where *n* is an auto-incremented number. Here you can specify the file name to use instead of the default name. |
| Output format | The output format can be selected from the list (CSV, XLS, XLSX, XLSM, DOC, DOCX, TXT with or without column headers, or XML). The output format is also overridden if selecting a template (*Contents* field) that uses a different file format.<br>By default, the data is extracted in an MS Excel file. |
| Contents | The output file can be based on a specific template that determines the content of the final extract as well as the file format (any eventual selection in the *Output format* field is overridden by the template). |
| Display template selection | When a template (content) type is selected but there are multiple template files for the selected type, the system will automatically select the first template it finds in the language that matches the selected output language (see below) or the connected user's language.<br>When enabled, the option *Show template choice* will force the display of a selector prompting users to choose the template they want to use each they run the query. |
| CSV separator | By default, the character used as column separator in CSV files or plain text is the semicolon (";"). This field allows you to force the use of another separator. |
| Text encoding | Encodage de sortie à utiliser pour les formats CSV et TXT |
| Use UTF-8 BOM identifier | This is enabled by default and applies when the selected text encoding is Unicode UTF-8. Disabling this checkbox allows you to discard the otherwise included BOM identifier from the generated file. |
| Timeout | This allows you to modify the maximum execution time of the query.<br>Access to this drop-down list is controlled by the authorization *QUE – Modify the query timeout* (auth_que_query_timeout).<br>Caution, the value you set for this parameter is sensitive: setting the value too high can render the timeout mechanism pointless; however, this mechanism serves the purpose of interrupting queries whose execution time is too long (which can freeze the application). |
| Transaction isolation mode | Specifies the lock type and version control behavior for the Transact-SQL instructions lines that are sent during a connection to SQL Server<br>The following modes are available:<br>▪ **No lock mode**: (default) same as SQL Server's "read uncommitted" mode.<br>One transaction may see not-yet-committed changes made by other transactions (no lock, partial data integrity). This is the most efficient mode (recommended).<br>▪ **Snapshot Mode**: Same as SQL Server's "snapshot" mode.<br>Changes made by other transactions after the active transaction started are not visible to said transaction. The transaction appears to operate on a personal snapshot of the database, taken at the start of the transaction (no lock, data integrity maintained). Medium performance. This mode requires a specific configuration of the database.<br>▪ **Locking mode**: Same as SQL Server's "read committed" mode.<br>One transaction may not see not-yet-committed changes made by other transactions, which prevents dirty reads (update lock full data integrity). This is the least efficient mode (not recommended for large queries). |
| Max rows | This allows you to specify the maximum number of rows that may be returned by the query. |

| *Field* | *Description* |
|---|---|
| | Default value = 65 000, configurable<br><br>Access authorization: *QUE–Modify query max rows* (auth_que_query_max_rows) |
| Header in extract | Includes column headers in query result extracts. |
| CSV/TXT double quote delimited | If this option is selected, the values are delimited by double quotes in plain text (TXT) files and in CSV files. In the case of CSV files, the specified separator still delimits each field, and the value within each field is between double quotes. |
| Create a log | Enables the saving of the query execution log in the journal. See *Viewing the logs*, page 34. |
| Export language | By default, the results are extracted and presented in the language of the logged-in user (*user_language*). You can force the use of a particular language by selecting from the list. |
| Sub-query | Allows you to select one or more sub-queries, which will be run with the main query. The main query and its sub-queries are linked by joins.<br><br>Joins between the main query and subqueries allow you to generate dynamic lists in MS Word, in workflow notifications or contract clauses (in the form of enumerations, numbered or bulleted lists, or tables). For instance, in a notification triggered by the invoice workflow, you could run a query on the Invoice table to pull information through merge fields (main query); If you link this main query to a sub-query on the Invoice line table, you would now be able to display dynamically the list of the invoice's lines.<br><br>When there is a join between a query and a subquery, the data from the subquery is automatically filtered based on the main query (similar to the results that an SQL INNER JOIN would produce).<br><br>For further information on:<br><br>▪ How to link a sub-query to the main query, see p.31<br><br>▪ How to use queries and sub-queries to insert merge fields and dynamic lists in a notification, please refer to the Administrator manual (Notifications and broadcasts) |

## Defining the characteristics of the output file

By default, the query results are retrieved in the language of the logged-in user, in a file in MS Excel format named "*Query* (auto-incremented number)."

### File name

You can modify the output file name by entering the name of the file to be used in the *Output file name* field.

It is possible to append the file name with a timestamp and/or a GUID.

| ***Format option*** | ***Syntax (Caution: case sensitive!)*** |
|---|---|
| GUID | [guid] |
| DATE | [date(date format)]<br>The date format is fully customizable using the custom date and time format specifiers from .NET Framework. Examples:<br>[date(yyyy/MM/dd)] 2015_01_28<br>[date(dd/MM/yyyy)] 28_01_2015<br>[date(MM/dd/yyyy)] 01-28-2015<br>[date(yyyy/MM/d hh:mm:ss tt)] 2015_01_28_03_55_26_PM<br>[date(yyyy/MM/d HH:mm:ss)] 2015_01_28_15_55_26<br>[date(yyyyMMd HHmmss)] 20150128_155526<br>To see more options and obtain more information on custom date and time format specifiers:<br>https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings |

### File format

You can choose the output file format from the *Output format* list, such as MS Excel (csv, xls, xslx, xslm), Plain text (txt), MS Word (doc, docx) or XML.

The query results can also be formatted by applying a template.

To do this, use *Contents* field to select a template, or click *See All* to display the full list in a new window; from the *Template Selector* window select the template by selecting the corresponding checkbox, then close the window.

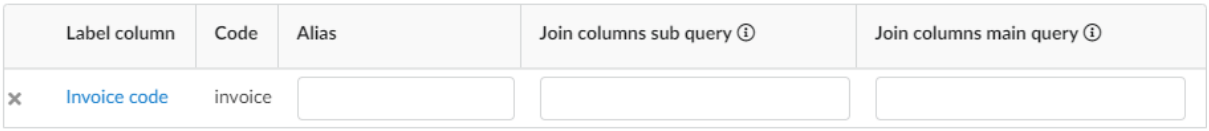
### Language

By default, the results are extracted in the language of the logged-in user (*user_language*), but it is possible to override the output language used for the extracted data by selecting a different one from the *Export language* drop-down list.

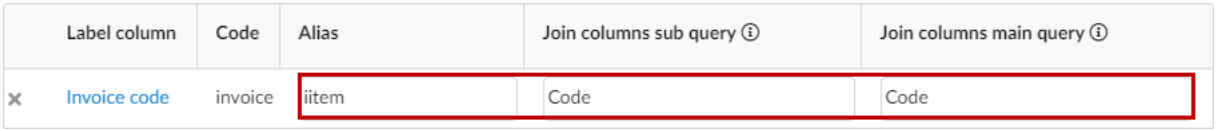
### Linking a sub-query to a query

The sub-query must be created beforehand. It must include a column that will allow you to create a join with the main query.

1. Access the *Publish* tab of the main query.
2. Use the *Sub-query* field to select the desired sub-query (click See All to display the full list in a new window, then select the corresponding checkboxes and close the window), then click the **Save** button.

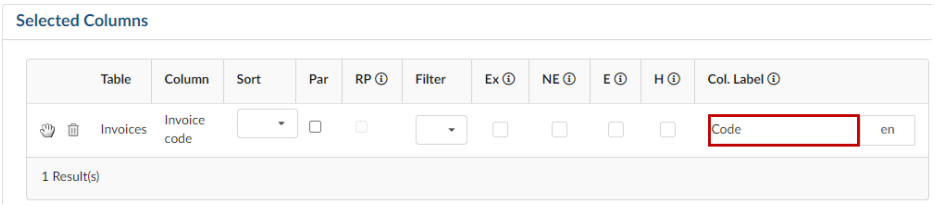



3. Create the join between the main query and the sub-query.




To do so:

- In the *Alias* column, declare the alias of the sub-query's main table (as defined in the t_bas_tab_desc table you can access via the *Admin > Browse Data* menu)
- In the columns *Join columns sub-query* and *Join columns main query*, declare the alias of join columns, separated by ";" if more than one (column aliases are defined in the t_bas_col_desc table or on the *Query Manage* tab, *Selected columns* area, *Col. Label* column, see screenshot below)




4. Click the **Save** button.

# Preview and execution

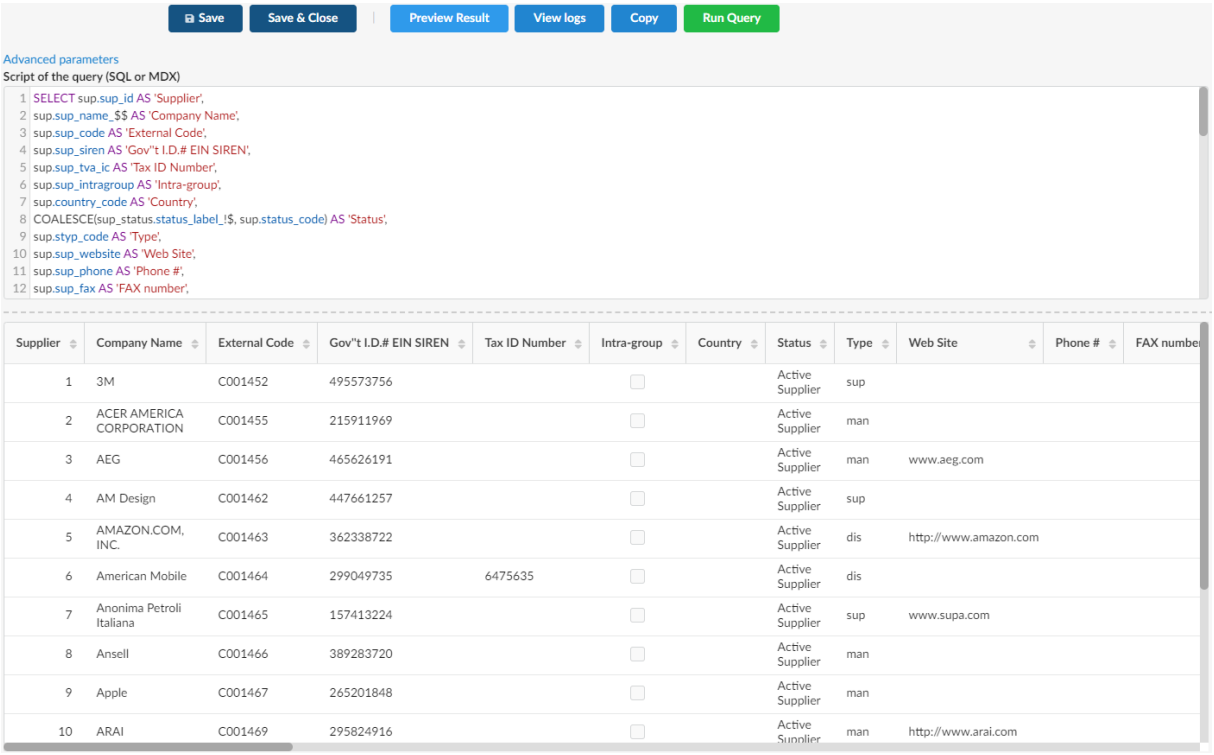
## Prerequisite

To be able to display or extract the results of a query, you must save it first. To do this, you must at least specify its code and title.

## Previewing the raw result of the query

The preview function allows you to view the query result in the query editor (*Query* tab), below the *Script of the query (SQL or MDX)* field. This provides a direct on-screen preview and allows you to edit the query code if necessary.

To preview the result of a query, click the **Preview Result** button in the query record (Action bar).



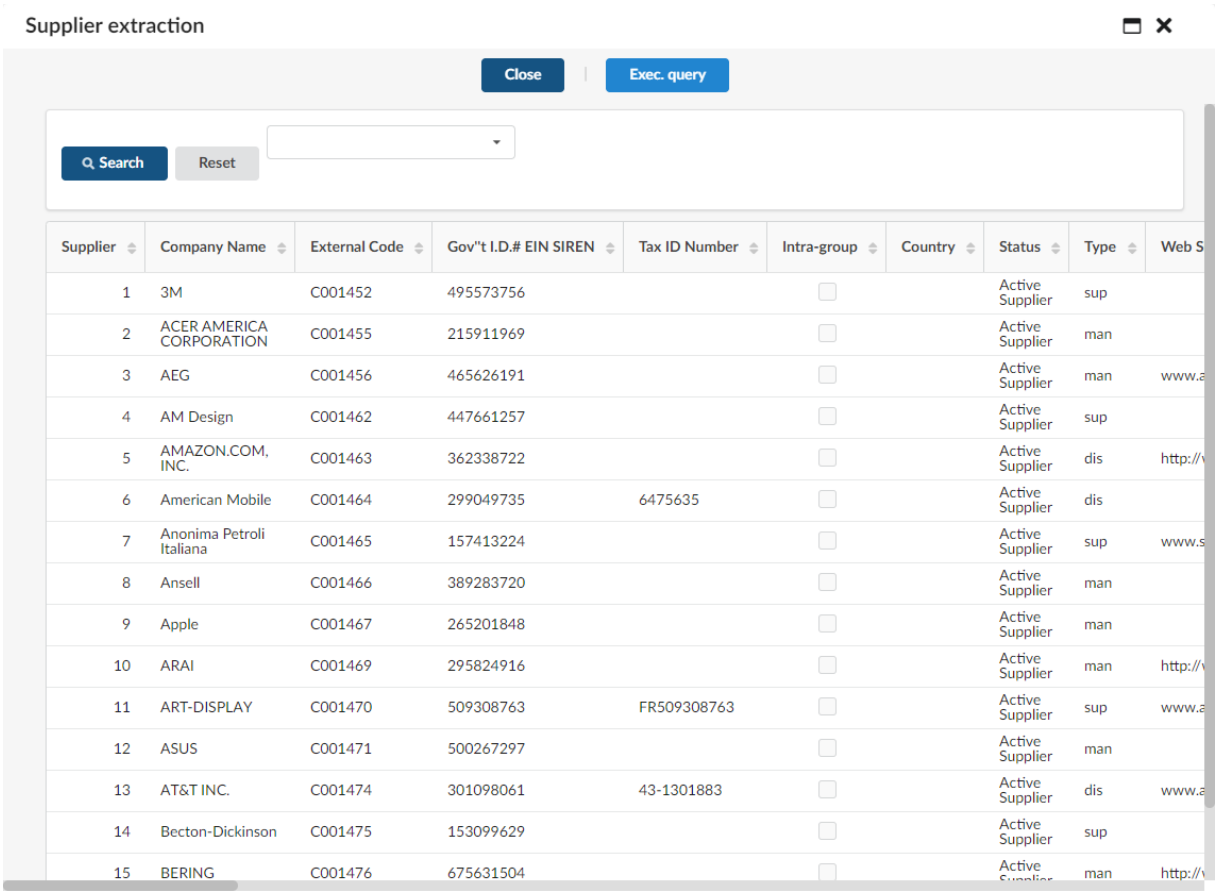

| Supplier | Company Name | External Code | Gov"t I.D.# EIN SIREN | Tax ID Number | Intra-group | Country | Status | Type | Web Site | Phone # | FAX numbe |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3M | C001452 | 495573756 | | | | Active Supplier | sup | | | |
| 2 | ACER AMERICA CORPORATION | C001455 | 215911969 | | | | Active Supplier | man | | | |
| 3 | AEG | C001456 | 465626191 | | | | Active Supplier | man | www.aeg.com | | |
| 4 | AM Design | C001462 | 447661257 | | | | Active Supplier | sup | | | |
| 5 | AMAZON.COM, INC. | C001463 | 362338722 | | | | Active Supplier | dis | http://www.amazon.com | | |
| 6 | American Mobile | C001464 | 299049735 | 6475635 | | | Active Supplier | dis | | | |
| 7 | Anonima Petroli Italiana | C001465 | 157413224 | | | | Active Supplier | sup | www.supa.com | | |
| 8 | Ansell | C001466 | 389283720 | | | | Active Supplier | man | | | |
| 9 | Apple | C001467 | 265201848 | | | | Active Supplier | man | | | |
| 10 | ARAI | C001469 | 295824916 | | | | Active Supplier | man | http://www.arai.com | | |

The preview takes the sorting order into account, as well as column renaming and the filters before execution, however, during query execution, filters (defined via the function Parameter) are ignored.

## Showing the results as a search page

In the list of queries, click the *Preview* icon to open the preview results in a pop-up window.

In this view of the results, the query name is used as the page title and filters at runtime are proposed in the search filters area at the top of the page.



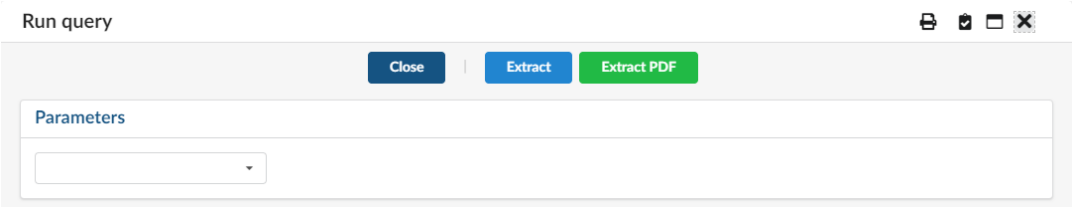

| Supplier | Company Name | External Code | Gov"t I.D.# EIN SIREN | Tax ID Number | Intra-group | Country | Status | Type | Web S |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3M | C001452 | 495573756 | | | | Active Supplier | sup | |
| 2 | ACER AMERICA CORPORATION | C001455 | 215911969 | | | | Active Supplier | man | |
| 3 | AEG | C001456 | 465626191 | | | | Active Supplier | man | www.a |
| 4 | AM Design | C001462 | 447661257 | | | | Active Supplier | sup | |
| 5 | AMAZON.COM, INC. | C001463 | 362338722 | | | | Active Supplier | dis | http://v |
| 6 | American Mobile | C001464 | 299049735 | 6475635 | | | Active Supplier | dis | |
| 7 | Anonima Petroli Italiana | C001465 | 157413224 | | | | Active Supplier | sup | www.s |
| 8 | Ansell | C001466 | 389283720 | | | | Active Supplier | man | |
| 9 | Apple | C001467 | 265201848 | | | | Active Supplier | man | |
| 10 | ARAI | C001469 | 295824916 | | | | Active Supplier | man | http://v |
| 11 | ART-DISPLAY | C001470 | 509308763 | FR509308763 | | | Active Supplier | sup | www.a |
| 12 | ASUS | C001471 | 500267297 | | | | Active Supplier | man | |
| 13 | AT&T INC. | C001474 | 301098061 | 43-1301883 | | | Active Supplier | dis | www.a |
| 14 | Becton-Dickinson | C001475 | 153099629 | | | | Active Supplier | sup | |
| 15 | BERING | C001476 | 675631504 | | | | Active Supplier | man | http://v |

## Running the query

To run the query, you can:

- Either click the **Run Query** button in the Query record
- Or click the corresponding the *Extract* icon in the list of queries

If the query includes filters at runtime (**Par** function), a parameter input window will be displayed.



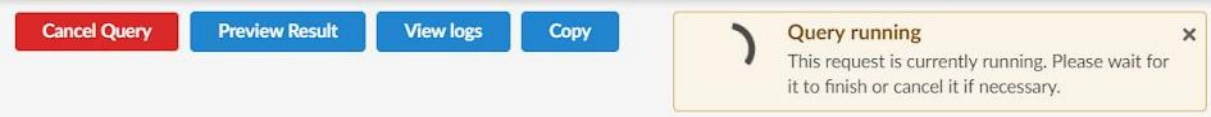

The input of filters is optional.

Click the **Extract** button to extract the results in the output format selected on the *Publish* tab of the query; or, click on **Extract PDF** button to generate a PDF output of extract results.

From version 8.164, you can interrupt a running query by clicking the **Cancel Query** button displayed in place of the corresponding **Run Query** or **Extract** button.

**Note:** If running a query from the Browse Queries page (*Analytics* > *Browse Queries* menu), you can cancel the execution by clicking the corresponding button in the *Extract Preview* column.

A tooltip informs you of the execution status:




Note that the same query cannot be launched more than once.

Depending on the situation, the following messages can be displayed in the tooltip:

- *Query running*: Normal operation. Wait for execution completion or, if needed, you can cancel the execution.
- *Query already running*: If you try to run a concurrent instance (e.g.: from another window), the application displays informs you that the same query is already being processed. Wait until the execution completes.
- *Too many queries are currently executing on this server for your login*: If you have reached the maximum per-user query parallel execution value, the application indicates you that no more queries can be launched (see *Query maximum parallel execution*, page 49). You can cancel the execution and wait for any running query to complete before trying again.

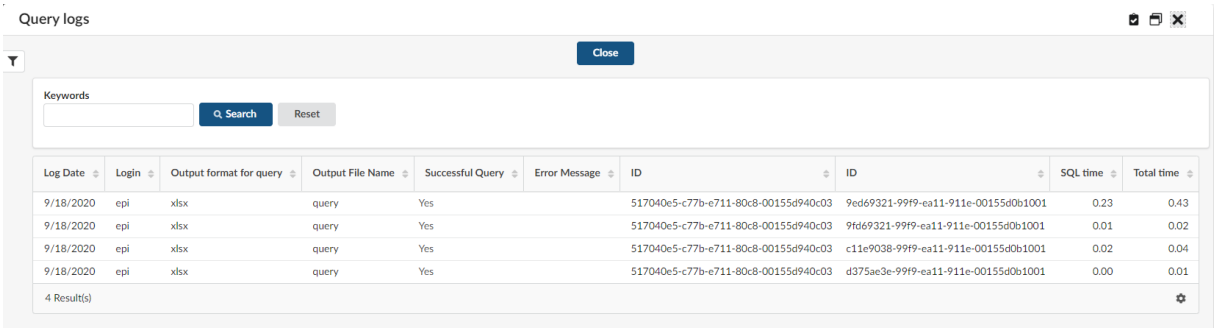
## Viewing the logs

To enable logging of query runtime data, select the *Create a log* checkbox on the *Publish* tab prior to running your query.

To access this information after having run the query, click the **View logs** button in the action bar.

Logs record the following times:

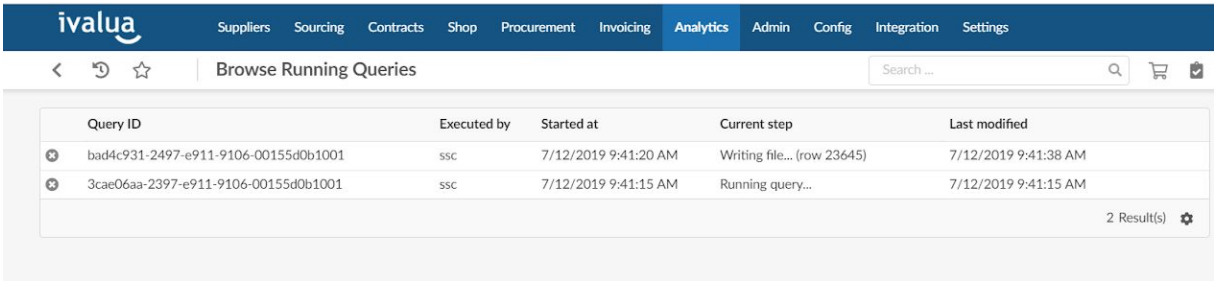
- SQL time (query run time)
- Rendering time (document construction)
- Total time (SQL + rendering)



## Accessing and Managing Running Queries

You can review the running queries and stop them if needed (administrators can see all the queries that belong to any other user).

Select the *Analysis* > *Browse Running Queries* menu.



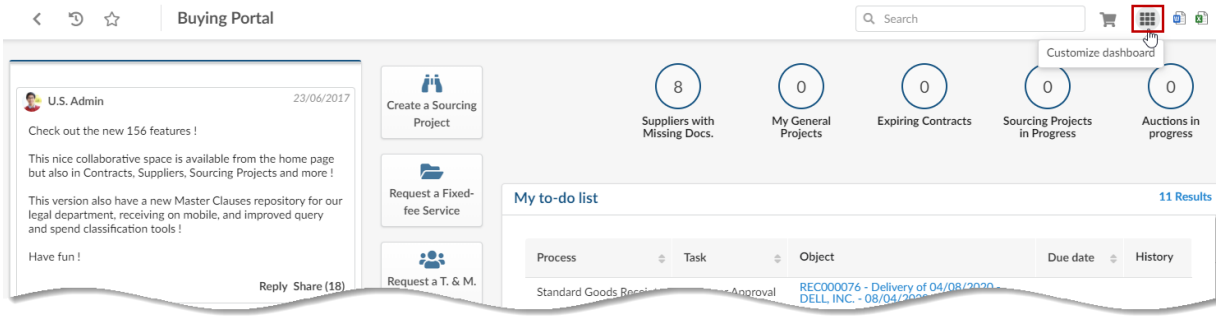

From this page, you can stop the execution of a given query by clicking the corresponding icon next to the *Query ID*.

# DISPLAYING THE RESULTS OF A QUERY

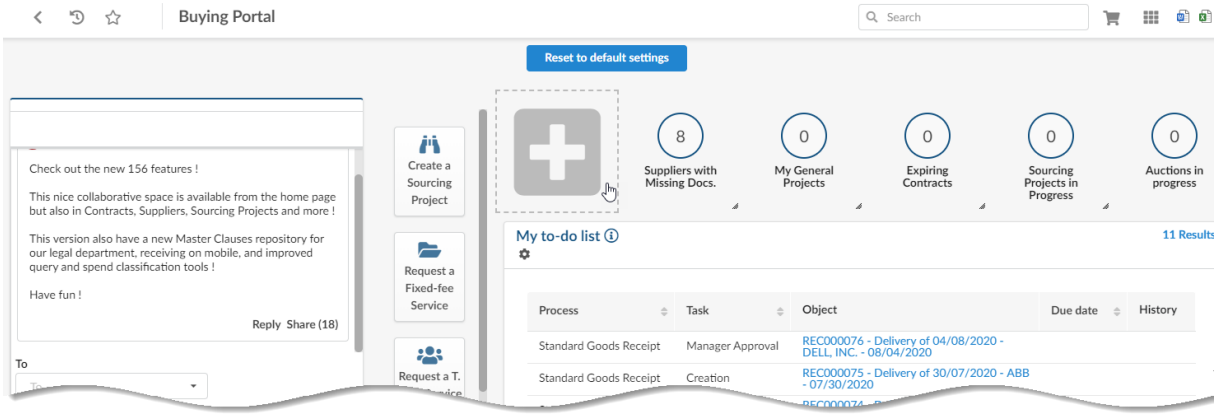
# Inserting a query on the home page (webpart)

### Accessing the Configuration of the content of a default home page

Display the Home page and click on the *Customize Dashboard* icon located in the navigation header.




The homepage is now in customization mode:

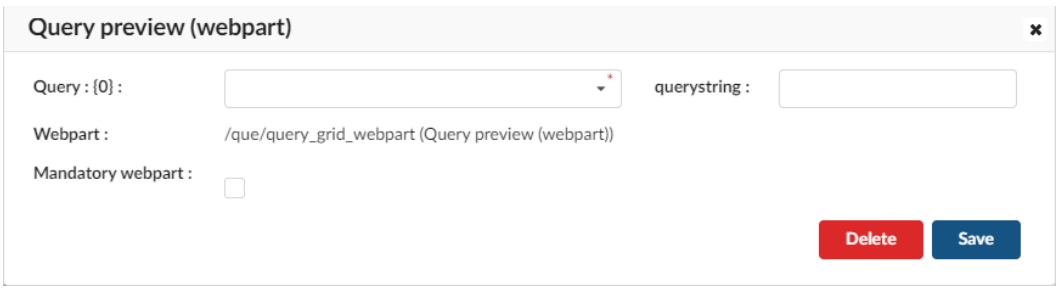



.

In this mode, you can add new webparts by clicking the empty areas on the page. Locations that allow placing a webpart are indicated by a large "**+**" on mouseover (as shown above).

Existing webparts can be moved. Drag and drop the webparts to move them throughout the page grid.

Each webpart can be configured by clicking on the *Parameters* icon below its title. The parameters open in a pop-up window. You can also delete the webpart from this window.




In customization mode, the following buttons are available in the action bar:

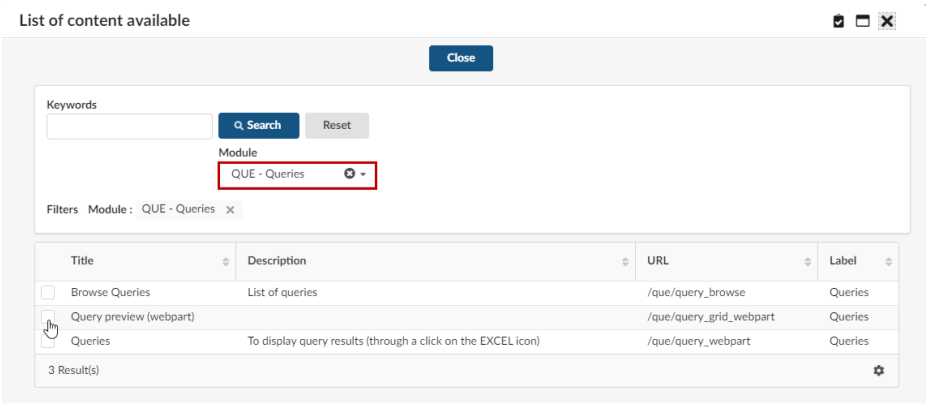
- *Reset to default settings*: deletes all custom webparts and resets the page to its default layout.
- *Set as global settings*: sets the page webparts and layout as the default for all users.

## Adding content

To add a query webpart on the homepage, proceed as follows:

1. Click the **+** button (you may have to scroll down below existing webparts to see it).

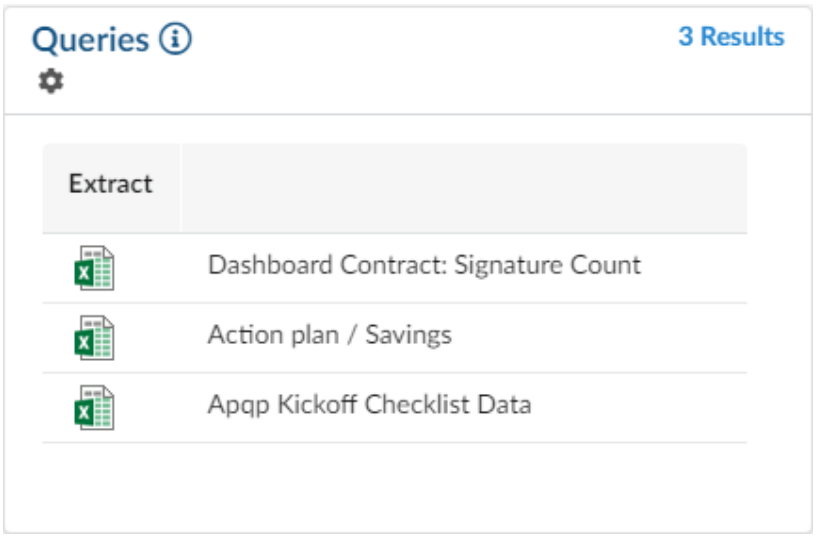
2. The list of available webparts and content is displayed. Filter the list by selecting the *QUE – Queries* option in the *Module* search field in order to display Query-related webparts only.



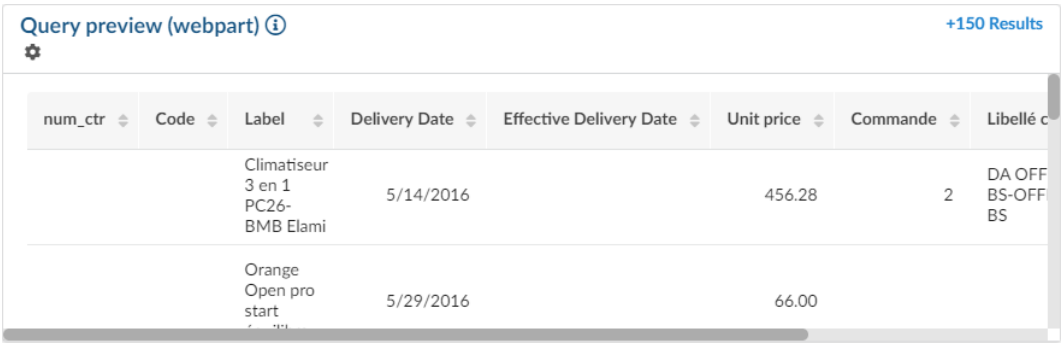

3. Select the checkbox that corresponds to the content you wish to include as a webpart on the homepage.

The first two content types below can insert Query reports as webparts on the homepage:

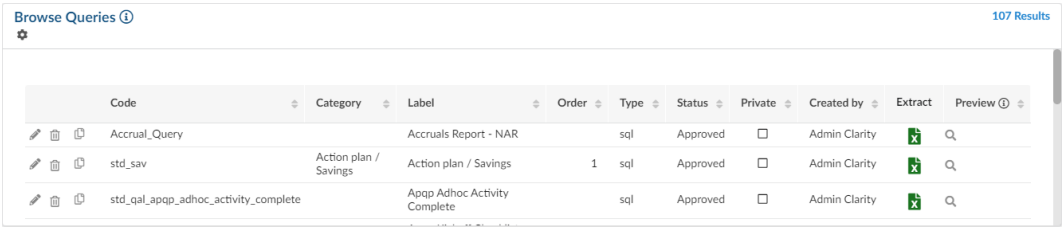
**Queries** Insert a link to open the query result in a file



**Query preview (webpart)** Directly displays the query results in the webpart

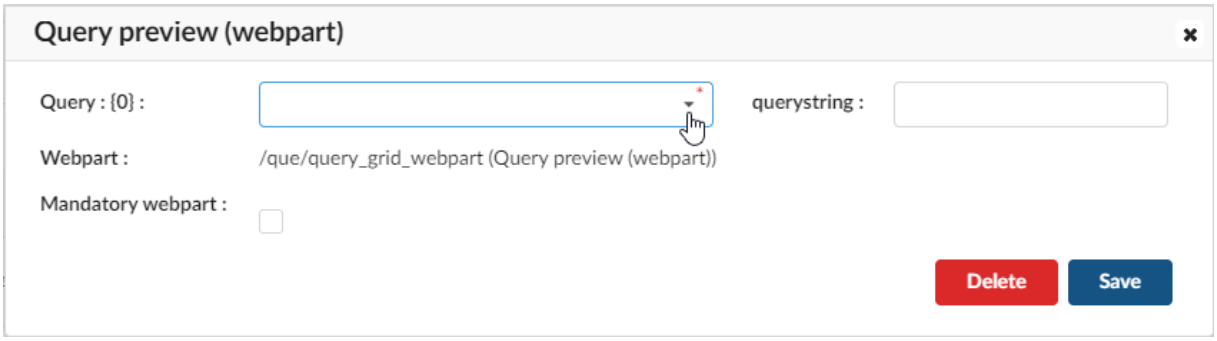



**Browse queries**




## Selecting the query to be displayed in the webpart

Click on the icon of the webpart in order to access its parameters.




Open the *Query* field list and select the query you wish to display. You can access the full list by clicking *See all*. The list of available queries is displayed in a new window; select the query by selecting the corresponding checkbox.

Click the **Save** button to add the query as a webpart.

## Quitting the Configuration mode of the Home page

When you are satisfied with the choice and arrangement of the contents, click *Customize Dashboard* icon.

All the changes are saved and the pages goes back to normal operation mode.

# CONFIGURATION

# Table Management

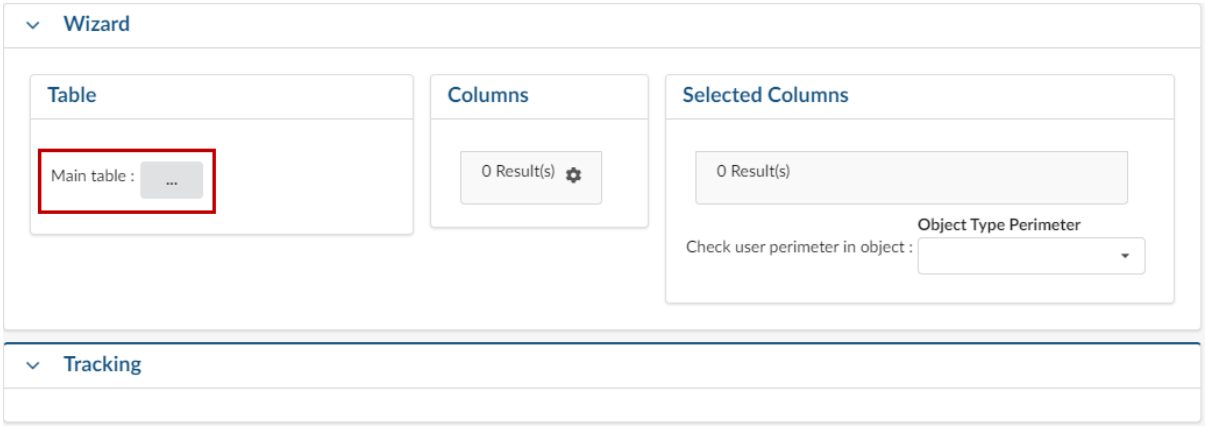
The basic tables and columns that can be subject to queries are defined in the BAS – Table (`t_bas_tab_desc`) and BAS – Column (`t_bas_col_desc`) tables.

You can access these tables from the *Admin* > *Browse data* menu.

### BAS – Description of the tables of data model

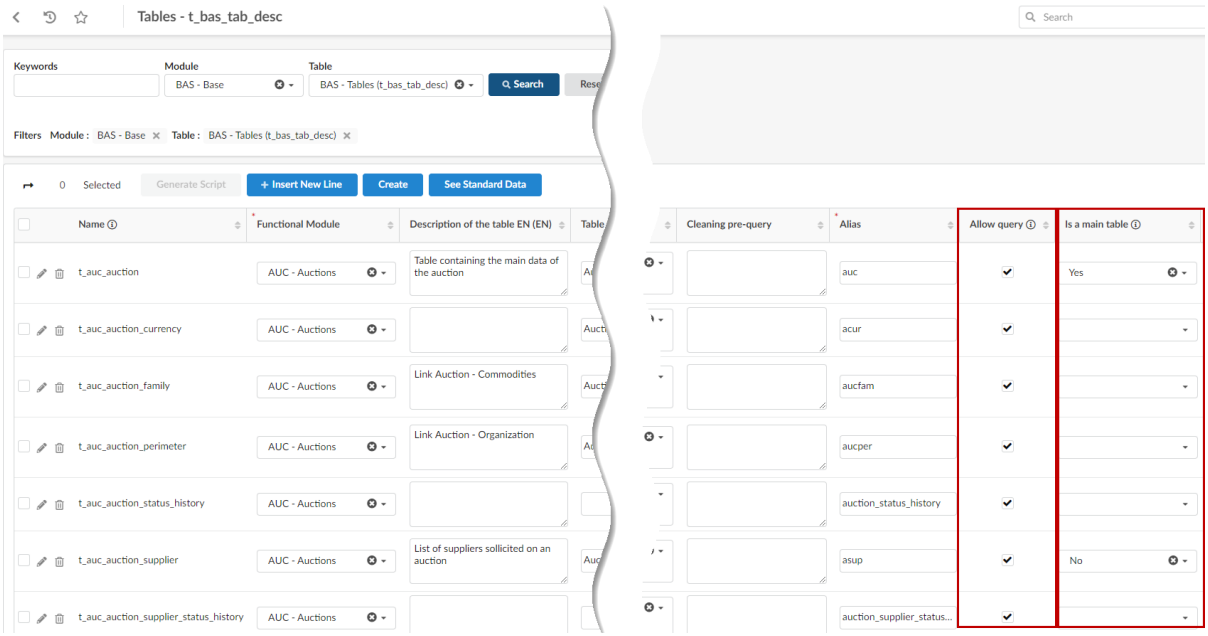
Access the `t_bas_tab_desc` table content (use the search fields for easier access).

**Is Main table**: determines which tables are available as a starting point in building a query, i.e. those that are selectable as the main query table.




To set a table as a main table to use in the query maker (and the Wizard), set the value of the *Is main table* field to *Yes* for the corresponding table.

**Allow query**: determines whether the table will be accessible in the query builder. Select this checkbox for the corresponding table.

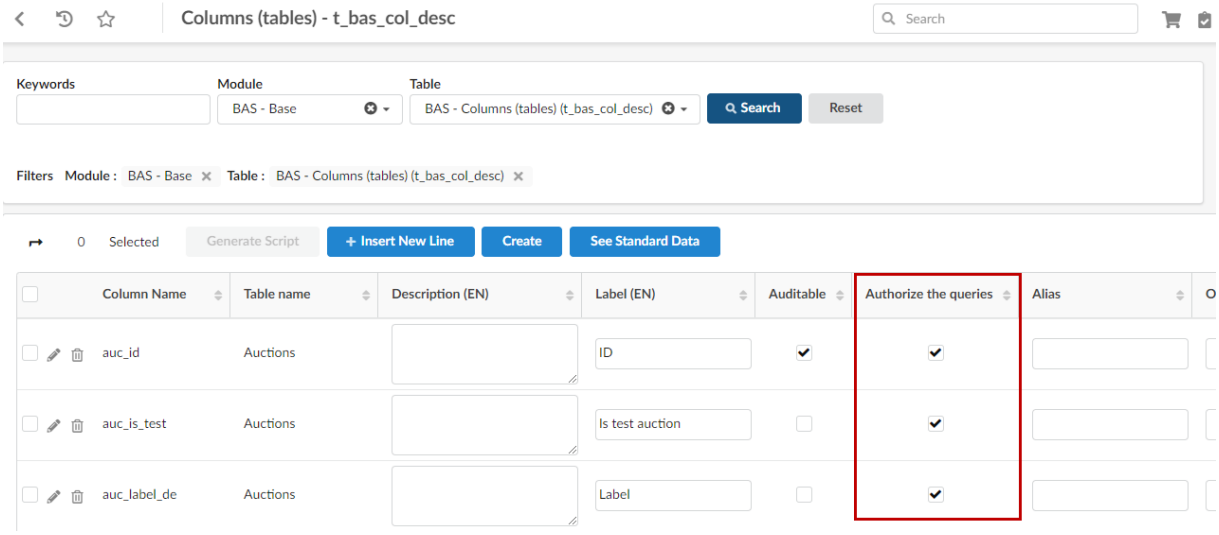



### BAS – Description of table columns

Access the `t_bas_col_desc` table content (use the search fields for easier access).

**Authorize the queries**: determines whether the column will be accessible in the query maker.

Select this checkbox for the corresponding column in the list.



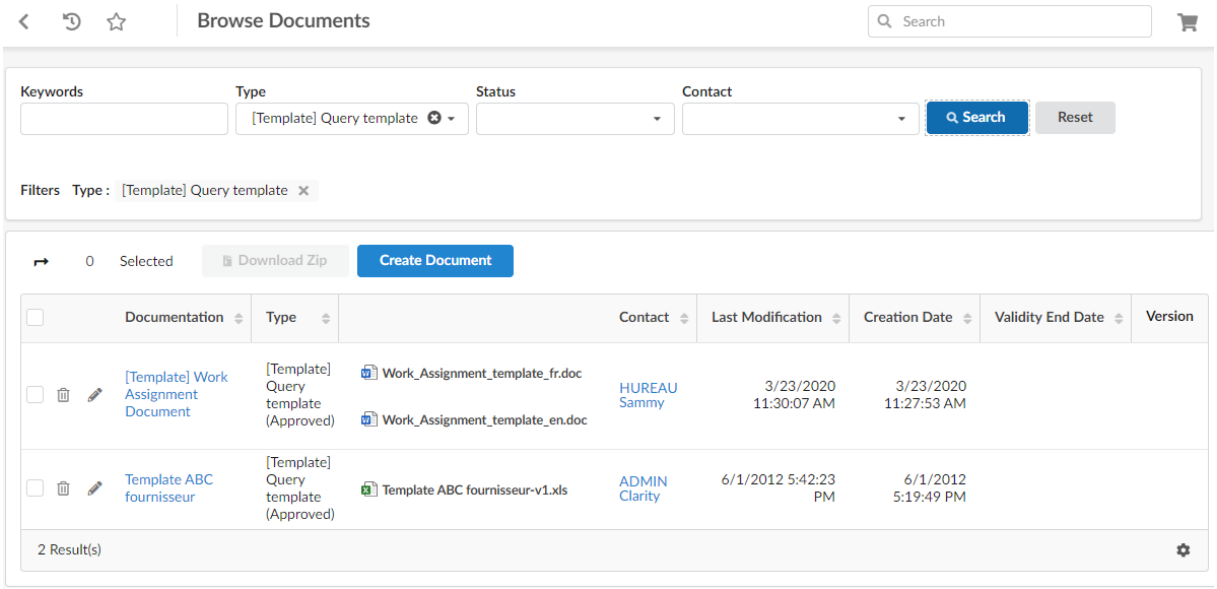

# Managing query templates

## Principle

The results of a query can be formatted by applying a selected template on the *Publish* tab (*Content*).

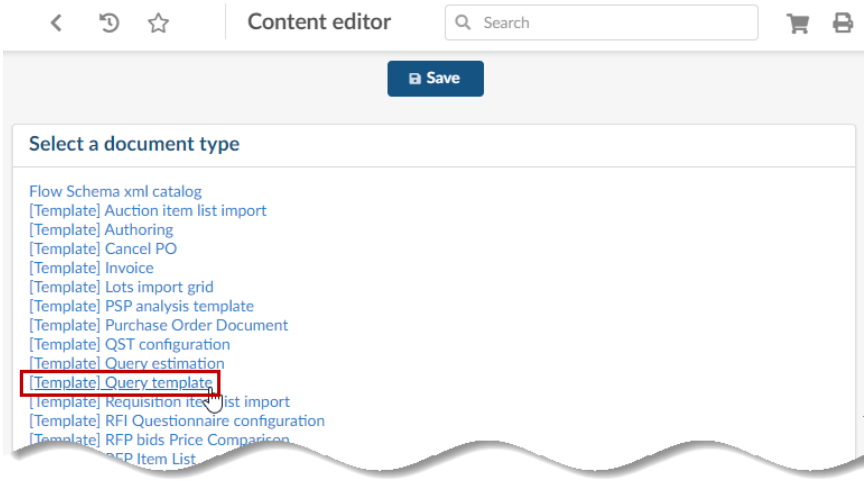
The templates that are available to be selected are contents of the *Query template* type. They must be created prior to query creation in order to be usable in the query builder.

**Note:** You can filter the display by selecting the *Query Template* type in the *Type* search filter field. If this filter is used, clicking the *Create Document* button directly selects the query template type (bypasses the document type selection page).

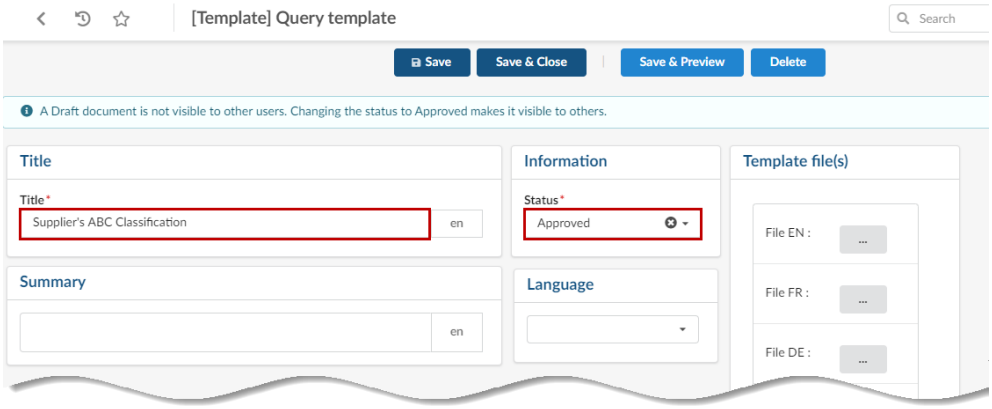


## Creating/Modifying a *Query template* type of content

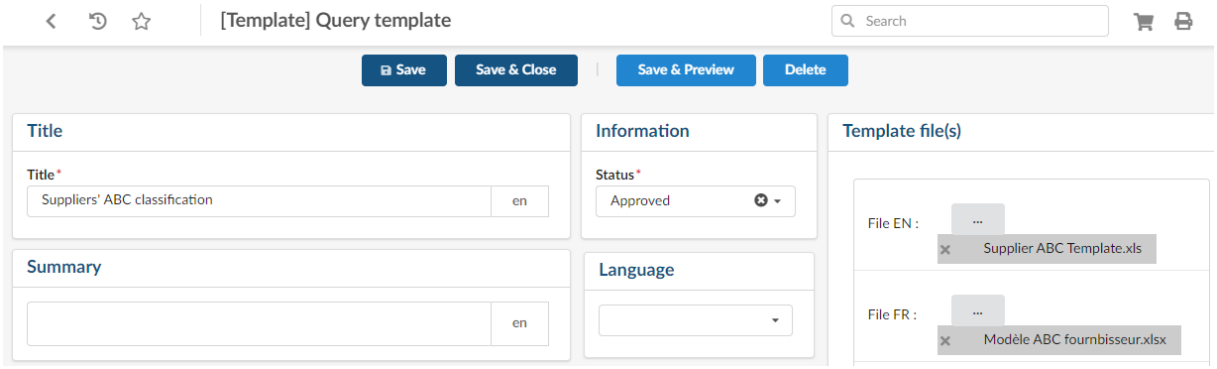
1. Access the *Admin* > *Browse Documents* menu.
2. Click on the **Create Document** button. The *Edit content* page will be displayed.
3. Select the *Query template* content type.




4. Specify the template title and select the *Approved* status.



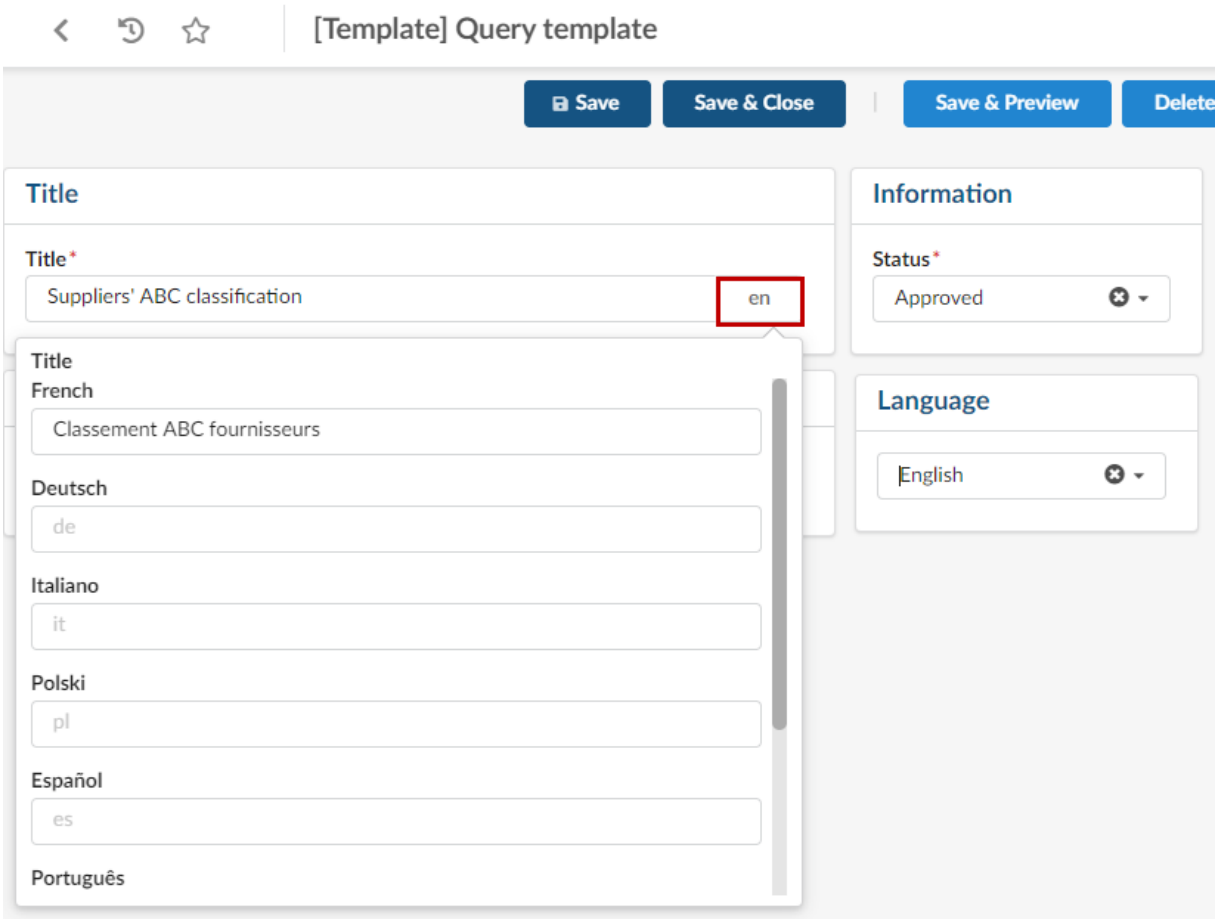

5. Add the template file as an attachment. Click the ... button linked to the file language in the *Template file(s)* area, select the template, then in the *Template* window, click the **Click or Drag to add file** link to select a template file from your computer. Alternatively, you can select an existing file from the *Template for Export* list.
6. Click the **Save** button.



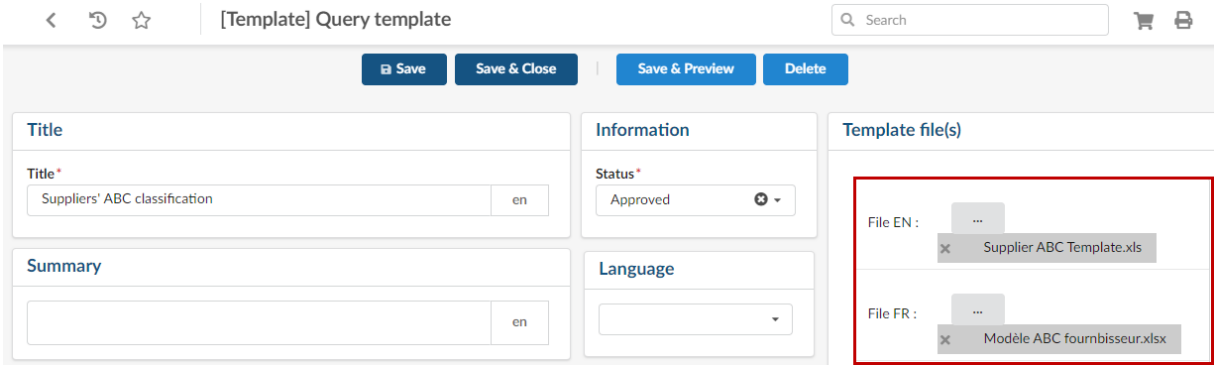
## Multilingual templates

By default, the query results are extracted in the language of the logged-in user (Output language: *user_language*). You can override the extraction language used by the template by selecting a different one from the *Language* dropdown list.

The template title can be translated in the other languages supported by the application. Click the language indicator in the *Title* field to open the translation tooltip. The content of the *Summary* field can also be translated by using the tooltip.




When a document template is used, it is possible to manage a multilingual extraction by attaching the file in each language you want to support.



# Query maximum parallel execution

From version 8.164, query execution can be limited in order to avoid server overload and long-running queries.

This can be particularly useful if a given query needs to process large amounts of data, and that may therefore seem to be unresponsive. It prevents users from launching multiple instances of the same query repeatedly. The application also allows to cancel the execution of already running queries.

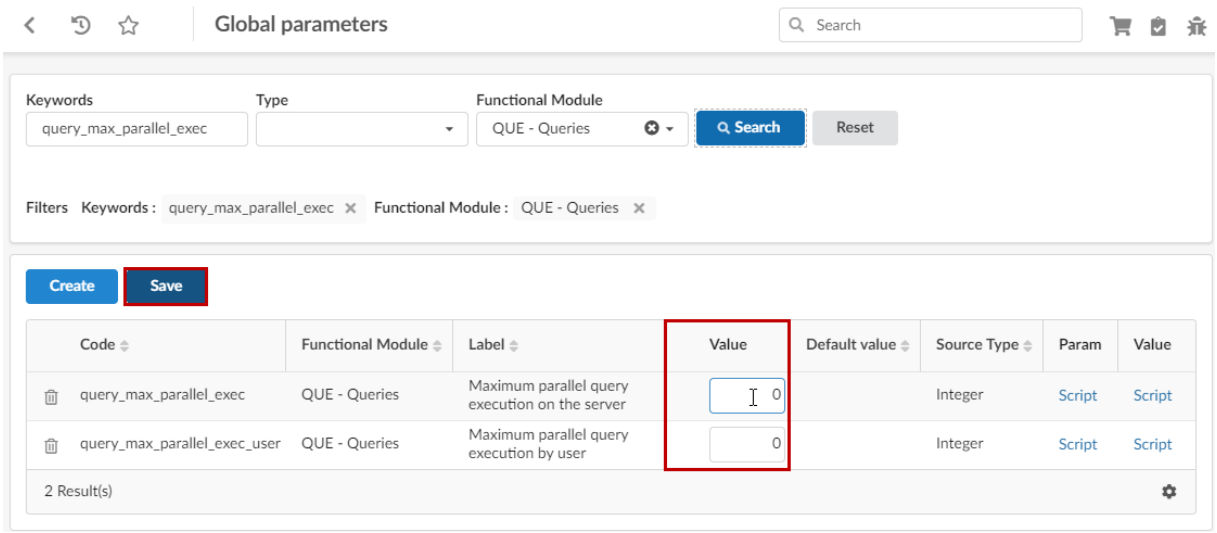
This configuration also reduces the risk of having poorly written or invalid queries (e.g.: unoptimized queries or code that leads the query processing engine to hang for long).

Query execution limitation prevents processing and memory overload by enforcing a fixed limit set by the administrator.

The defined value is the maximum number of queries that can run in parallel:

- Globally. The maximum total number of queries that can run in parallel on the server.
- Per-user. This limit defines the maximum number of queries individual users can run simultaneously.

Access the *Settings* > *Parameters* menu, then, use the *Keywords* field to search for the *query_max_parallel_exec* parameters for the *QUE – Queries* Functional Module.




The two following parameters are available:

- *query_max_parallel_exec*: Global limit. Allows to define the maximum total number of parallel query executions on the server (regardless of per-user limits).
- *query_max_parallel_exec_user*: allows to define the maximum per-user parallel query execution.

Specify a positive integer value for each field in the *Value* column.

A value of 0 (zero) means that there is no enforced limit.

Click the **Save** button to apply the changes.